

RESEARCH

Open Access



A novel queue management policy for delay-tolerant networks

Saeid Iranmanesh

Abstract

Delay-tolerant networks (DTNs) have attracted increasing attention from governments, academia and industries in recent years. They are designed to provide a communication channel that exploits the inherent mobility of trams, buses and cars. However, the resulting highly dynamic network suffers from frequent disconnections, thereby making node-to-node communications extremely challenging. Researchers have thus proposed many routing/forwarding strategies in order to achieve high delivery ratios and/or low latencies and/or low overheads. Their main idea is to have nodes store and carry information bundles until a forwarding opportunity arises. This, however, creates the following problems. Nodes may have short contacts and/or insufficient buffer space. Consequently, nodes need to determine (i) the delivery order of bundles at each forwarding opportunity and (ii) the bundles that should be dropped when their buffer is full. To this end, we propose an efficient scheduling and drop policy for use under *quota*-based protocols. In particular, we make use of the encounter rate of nodes and context information such as time to live, number of available replicas and maximum number of forwarded bundle replicas to derive a bundle's priority. Simulation results, over a service quality metric comprising of delivery, delay and overhead, show that the proposed policy achieves up to 80 % improvement when nodes have an infinite buffer and up to 35 % when nodes have a finite buffer over six popular queuing policies: Drop Oldest (DO), Last Input First Output (LIFO), First Input First Output (FIFO), Most Forwarded first (MOFO), LEast PRobable first (LEPR) and drop bundles with the greatest hop-count (HOP-COUNT).

Keywords: Delay-tolerant networks, Congestion, Drop policy, Scheduling policy, Buffer management

1 Introduction

In delay-tolerant networks (DTNs) [1], delay-insensitive data are propagated between nodes whenever they are within radio range of one another. Example applications include exchanges between cars in a city to provide traffic information, safety messages regarding accidents and events of interest to drivers in different areas. This in turn warns drivers to use an alternative route and thereby relieves traffic congestion at accident sites. A key characteristic of DTNs is the lack of contemporaneous paths between any source and destination nodes. Hence, other nodes, i.e. trams, buses, cars and people, are involved to act as relays to help forward bundles/messages. However, these nodes may have intermittent connectivity because they are highly mobile or have a short contact period. As an example, consider two vehicles travelling in opposite directions at a speed of 20 m/s and have a radio

range of 40 m. The link between the two vehicles will last for $40/20 = 2$ s assuming negligible channel discovery time. A study on vehicular networks [2] shows that the duration of contacts between cars using IEEE 802.11 g crossing at 5.5 m/s is about 40 s, at 11.11 m/s, it is about 15 s and at 16.66 m/s, it is about 11 s.

DTN routing protocols can be classified into two groups based on the number of bundle replications [3, 4]: (i) flooding and (ii) quota. Flooding-based protocols send a replica of each bundle to any encountered nodes, whereas quota-based protocols restrict the number of replicas. In fact, unlike flooding-based routing protocols, the number of replicas in quota-based routing protocols is not dependent on the number of encounters and dictated by the specific policy of the protocol [5]. Flooding-based protocols do not require any knowledge of the network topology [5–7]. Despite their robust delivery ratio and low delay, flooding-based protocols have higher energy usage, bandwidth and buffer space consumption [7–9]. However, the buffer size of devices may be limited, which may lead

Correspondence: si175@uowmail.edu.au
School of Electrical, Computer and Telecommunication Engineering,
University of Wollongong, Wollongong, NSW, Australia

to bundle loss and low delivery ratios, especially during high traffic loads [5, 6, 10]. In contrast, quota-based protocols employ a limited number of replicas, which improve network resource usage [11]. This means, under quota protocols, if senders forward all replicas of a bundle to encountered vehicles, they are no longer allowed to replicate the said bundle. In fact, quota-based protocols have been proven to achieve a reasonable trade-off between routing performance and resource consumption [12]. However, these routing protocols suffer from comparatively lower delivery ratios even though they are resource friendly [13]. Moreover, a fixed number of replicas for bundle replication lack the flexibility to react to any changes in resource capacity [14].

As a critical consideration, a limited bandwidth and/or the insufficient duration of contacts may cause nodes to not be able to exchange all their bundles. Recall that the duration of contacts is affected by the speed of nodes. In such case, a strategy is required to manage the buffer in the case of forwarding and dropping. For example, assume that two nodes are in contact for 10 s and they have 20 bundles to forward to each other. Also, assume that the bandwidth capacity is 100 KBps, meaning that each node can forward 10 out of 20 bundles upon the contact. Hence, a queue policy makes the 20 bundles sorted based on the value of an objective function, e.g. delivery probability, delay. Then, bundles with higher priority are forwarded first. However, an inappropriate forwarding strategy may result in the system with a low delivery ratio and/or large delays. This is because bundles with a high delivery through a contact may not have the chance of being forwarded. Queue management is also needed when nodes experience buffer overflow. In this case, nodes with a full buffer need to provide room for incoming bundles. For this reason, they sort their buffered bundles based on the value of an objective function and drop a number of buffered bundles and replace them with incoming bundles. As a result, it is very important that nodes keep bundles in their buffer for a whilst until an appropriate forwarding opportunity arises. However, an inefficient drop policy may drop those bundles before meeting a good bundle carrier or even destination. Note that when bundles are sorted based on a metric, e.g. delivery probability, forward policy selects the bundle with the highest value whilst drop policy selects the bundles with the lowest value.

As a simple example of queue management, Fig. 1a shows that a bus and a motorbike have a 3-s contact period. The communication channel has a capacity of one bundle per second. The bus and motorbike have seven and five bundles, respectively, to exchange with each other. However, due to the short contact duration, they are unable to exchange all bundles. As a result, the bandwidth let only three bundles of each to be

forwarded (i.e. assuming the bandwidth is bidirectional). Hence, the bus and motorbike must decide which bundles to forward first. Also, notice that the bus's buffer is full. Therefore, the bus must determine which bundle(s) to drop; see Fig. 1b. In this example, the bus replaces three buffered bundles with the three received bundles based on a criterion. The motorbike also has two rooms available for the two received bundles from the bus and has to drop one of its buffered bundles to replace with the last received bundle. However, dropping bundles arbitrarily may cause delivery failure. This is because a bundle with a high delivery probability may be dropped. To this end, the bus and motorbike need to prioritize their respective bundles in order to decide which bundles they want to drop or forward with the goal of maximizing delivery ratio. Hence, it is important to have an efficient (i) bundle drop policy and (ii) forward scheduling policy to decide the best bundle(s) to exchange.

Current buffer management policies [15–17] use local knowledge, which can be a bundle's time to live (TTL), queue waiting time and a bundle's hop-count. For example, the authors of [17] apply the "shortest life time first" policy to the PROPHET [18] routing protocol. Other buffer management policies [19–21] use global knowledge. For example, the context information of all bundles and their replicas including their hop-count and number of replicas disseminated. This information is exchanged at each contact but may result in high control overheads and buffer consumption. Moreover, ensuring this context information up to date is challenging, given the large delays experienced by nodes.

As mentioned in Section 2, to date, all buffer management schemes are mostly targeted at flooding protocols. This is logical that under flooding-based protocols, congestion occurs more frequently than quota-based protocols. However, under flooding protocols, if a bundle is dropped, there is still a high probability for the bundle to reach its destination. On the other hand, in quota protocols, as each bundle has finite copies, once a replica is dropped, the delivery probability of the corresponding bundle reduces. In other words, no provisions are provided to replace a dropped replica in order to maintain a high delivery ratio [14]. In the worst-case scenario, source nodes may remove all replicas/copies of a bundle.

Given the said observations, in this paper, we propose an efficient scheduling and drop policy for *quota*-based routing protocols [3, 4]. Our policy, called Queue Management in Encounter-Based Routing Protocol (QM-EBRP), takes advantage of the following bundle and node information: number of available replicas, maximum number of forwarded replicas, time to live and rate of encounters. This information is encapsulated in a multi-objective utility function that is then used for dropping or forwarding bundles. The proposed multi-objective utility function

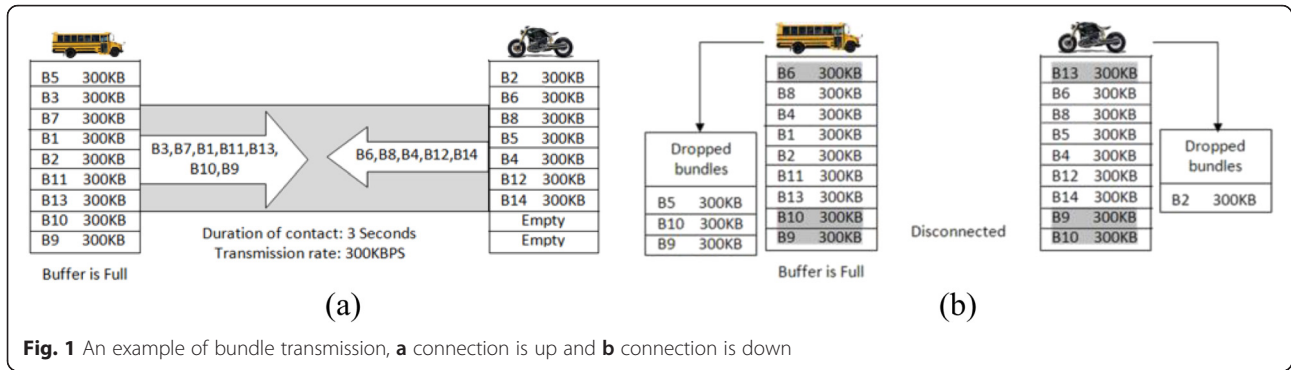


Fig. 1 An example of bundle transmission, **a** connection is up and **b** connection is down

incorporates two metrics: (i) delivery ratio and (ii) delay. The *delay* metric specifies how long it takes for a bundle to travel from a source to its destination, whilst *delivery ratio* is the total number of bundles that arrive at their intended destination successfully with respect to the total number of generated bundles. These metrics are formulated as delivery function and delay function. Then, in order to optimize the delivery ratio and delay concurrently, a multi-objective function is proposed to consider the rate of change of the said functions with respect to two parameters: *number of available replicas* and *time to live*. Note that these parameters highly impact on delivery ratio and delay. To this end, the objective function considers how fast a bundle reaches the maximum delivery rate and minimum delay. Hence, forwarding bundles with the highest rate of change will improve delivery ratio and delay. Our main contributions are as follows:

- Contrary to previous work [16, 19, 20, 22, 23], which requires global knowledge, QM-EBRP takes advantage of local information. An example of local information used by QM-EBRP is the maximum number of forwarded replicas. Although this is known as global information, QM-EBRP calculates this metric based on the remaining replicas that can be disseminated. This metric implies that if a large number of nodes receive a bundle's replica, the bundle will have a high probability of delivery. To date, many schemes [16, 19, 20, 24] *estimate* global information to avoid high energy and bandwidth consumption [21]. For example, Krifa et al. in [20] approximate the number of replicas and number of nodes (excluding sources) that have seen a bundle *i* since its creation based on the number of buffered bundles that were created before bundle *i*. These schemes are dependent on the dissemination rate of previous bundles and run the risk of using obsolete/inaccurate information. QM-EBRP also uses another local information, namely, the encounter rate of node, which current encounter-based protocols [3, 4, 18] compute by default locally. Hence, in the case where destination's mobility pattern is not fully predictable, forwarding bundles to nodes with a high encounter rate increases the chance of bundle delivery. This is because a node with a high encounter rate will have more contact opportunities to forward replicas than those with a low encounter rate.
- Previous works on buffer management such as [15–17, 19, 20] mostly focus on flooding protocols, where bundles can be replicated indefinitely. This causes more congestion than quota protocols. However, under quota-based protocols, where the number of replicas is finite, if congestion occurs, dropping a bundle may reduce the probability of delivery. In this respect, QM-EBRP is the first buffer management policy designed for quota-based routing protocols. Nodes that use a quota protocol are aware of the total number of replicas for each bundle. Hence, a node can easily obtain statistical information such as the number of available replicas in its buffer and maximum number of forwarded replicas it has disseminated throughout the network thus far.
- To maximize delivery ratio and minimize delays, we employ utility functions. Specifically, we use the gradient of these functions with respect to a bundle's remaining lifetime and number of replicas in order to maximize delivery ratio and reduce delay. Hence, if bundle *i* has a larger gradient as compared to bundle *j*, bundle *i* will have a higher delivery probability. Finally, through a standard normalization, we use the said gradient in a multi-objective utility function that aims to maximize delivery and minimize delay concurrently.
- In our experimental studies, over varying mobility models, we compare QM-EBRP to the following state-of-the-art queuing policies: Drop Oldest (DO), Last Input First Output (LIFO), First Input First Output (FIFO), Most Forwarded first (MOFO), LEast PRobable first (LEPR) and drop bundles with the greatest hop-count (HOP-COUNT). The results

show that under the shortest map-based mobility, when nodes have limited buffer space, i.e. each node has the capability of storing 5, 10, 20 and 40 bundles, QM-EBRP achieves up to 30 % and up to 80 % improvement when nodes have an infinite buffer. Also, under the working day movement model, QM-EBRP has up to 35 % improvement as compared to LIFO, MOFO, LEPR and HOP-COUNT when the buffer size of nodes is varied from 10 to 70 bundles in increments of 10 bundles. We also studied QM-EBRP and related policies over a model whereby nodes move randomly in a $2 \times 2 \text{ km}^2$ area. The results under random mobility show that QM-EBRP has up to 10 and 36 % improvement, respectively, as compared to DO and FIFO when each node has the capability of storing 10, 30, 50, 70, 90, 120, 150 and 200 bundles.

The rest of this paper is organized as follows. Section 2 describes the state of the art in buffer management and scheduling. Section 3 describes the system, and Section 4 proposes the queue management policy QM-EBRP. Section 5 evaluates QM-EBRP against well-known buffered management schemes, and finally, Section 6 concludes this paper.

2 Related works

We categorize current buffer management schemes into two groups: (a) *local knowledge schemes* [15, 17, 25–30] and (b) *global knowledge schemes* [16, 19–23, 31–35]. In the following sections, we review drop/forward policies in each category. Table 1 shows a taxonomy of all reviewed buffer management policies. Note that researchers have proposed many routing protocols for use in DTNs, challenging networks or vehicular networks. Interested readers are referred to [36, 37].

2.1 Local knowledge schemes

To date, past works have considered classical buffer management policies such as DO, Drop Random (DR), LIFO and FIFO for use in DTNs [38–41]. In DO, a node drops the bundle with the shortest TTL. The assumption is that a bundle with a short TTL implies it has been in the network for a long time and thus is likely to have been delivered. DR drops a bundle randomly. LIFO considers the arrival time of a bundle and drops the most recent bundle. In contrast, FIFO drops the bundle at the head of the queue, i.e. waited the longest. As long as the contact duration is sufficient to transmit all bundles, FIFO is a suitable policy. On the other hand, if the contact duration is limited, then FIFO fails because it does not provide any mechanism for preferential delivery or storing high priority messages. In [38], Dias et al. evaluated the impact of the

said policies on the performance of two routing protocols: epidemic [42] and Spray and Wait [43]. However, a bundle may have a small TTL but has a high probability to be delivered by a node. In this case, DO drops the bundle despite its high delivery probability.

In [15], Zhang et al. presented the impact of finite buffer and short contact duration when using the epidemic routing protocol [42] and evaluated drop policies such as drop-head (Drop Oldest), drop-tail and drop-head high priority. For the drop-head policy, when a node receives a new bundle and its buffer is full, the node drops the oldest bundle. Using drop-tail, when the buffer of a node is full, the node will not accept any bundle. As for the last policy, (i) if a source bundle, one that is transmitted by a source node, is sent to a node with a full buffer, the receiving node will first drop the oldest relayed bundle. Here, a “relayed bundle” is one forwarded by a non-source node. If there are no relayed bundles, the node drops the oldest source bundle, (ii) if a relayed bundle is sent to a node with a full buffer, the receiving node drops the oldest relayed bundle and if there is no relayed bundle, the new relayed bundle is not accepted.

Rashid et al. in [44] propose a drop policy which drops the stored bundle if its size is equal to or greater than the size of the incoming message. In another work [45], they drop bundles that have a size greater than or equal to the mean size of the node’s buffered bundles. Similarly, in [46], a bundle with the largest size will be dropped. In [47], a policy called Credit-Based Congestion Control mechanism is proposed that uses the age of a bundle as a heuristic to decide which bundle to be dropped when congestion occurs.

Recent work uses local knowledge in forward/drop policy. For example, Naves et al. [28] propose two drop policies: Less Probable Spray (LPS) and Least Recent Forward (LRF). In the former, a node uses the bundle delivery probability and estimates the number of replicas already disseminated to decide which bundle to drop. Hence, a node drops a bundle with the lowest delivery probability, only if it has disseminated the minimum number of replicas. This minimum is set according to network characteristics such as connectivity degree and inter-contact time. On the other hand, LRF, as its name implies, forwards the bundle that has not been forwarded over a certain period of time. In a similar work, Lindgren and Phanse [17] evaluated the following buffer management policies under the PROPHET [18] routing protocol: MOFO, most favourable first, DO, and LEPR. In the Most FORWARDED first policy, bundles that have been forwarded the most are dropped. In the most favourable first policy, the bundle with the highest delivery probability is dropped. The LEast PRobable first policy drops the bundle with the lowest delivery

Table 1 A classification of related works

Queue management policies	Drop Forward	Local global	Non-valid info	Consider delivered bundles	Predict bundle delivery	Utility	Consider finite replicas	Consider meeting rate
MV [20]	F	Local	No	No	Yes	DV	No	Yes
Rashid et al. [44]	D	Local	No	No	No	N/A	No	No
Rashid et al. [45]	D	Local	No	No	No	N/A	No	No
Rashid et al. [46]	D	Local	No	No	No	N/A	No	No
Rashid et al. [47]	D	Local	No	No	No	N/A	No	No
T-drop [21]	D	Local	No	No	No	N/A	No	No
Zhang et al. [11]	D	Local	No	No	No	N/A	No	No
Lindgren et al. [13]	D	Local	No	Yes	Yes	DV	No	Yes
Pan et al. [22]	D-F	Local	No	Yes	Yes	DV	No	Yes
LPS and LRF [23]	D	Local	No	No	No	DV	No	Yes
Fathima et al. [24]	D-F	Local	No	No	No	N/A	No	Yes
Rohner et al. [25]	F	Local	No	No	Yes	DV	No	Yes
Rashid et al. [39]	D	Local	No	No	No	DV	No	No
Rashid et al. [19]	D	Local	No	No	No	DV	No	No
Pan et al. [27]	D	Global	Yes	Yes	Yes	DV, DL	No	Yes
Yin et al. [26]	D-F	Global	Yes	Yes	Yes	DV + DL + OV	No	Yes
PREP [30]	D-F	Global	Yes	No	Yes	DV, DL	No	Yes
Yong et al. [18]	D	Global	Yes	Yes	Yes	DV	No	Yes
Dohyung et al. [12]	D	Global	Yes	Yes	No	DV	No	No
Krifa et al. [4]	D-F	Global	Yes	Yes	Yes	DV, DL	No	Yes
RAPID [17]	D-F	Global	Yes	Yes	Yes	DV	No	Yes
Krifa et al. [15]	D-F	Global	Yes	Yes	Yes	DV, DL	No	Yes
Elwhishi et al. [3]	D-F	Global	Yes	Yes	Yes	DV, DL	No	Yes
Liu et al. [28]	D-F	Global	Yes	Yes	Yes	DV	No	No
Shin et al. [29]	D-F	Global	Yes	Yes	Yes	DV	No	No
Yun et al. [49]	D	Global	Yes	No	No	N/A	No	No
QM-EBRP	D-F	Local	No	Yes	Yes	DV + DL	Yes	Yes

probability. The problem with the Most FORWARDED first policy is that it does not consider a bundle's lifetime, meaning a bundle with insufficient lifetime for delivery will not be dropped if the bundle has not been forwarded the most. Similar to [18], Rashid et al. [48] propose a buffer management policy called Message Drop Control Source Relay (MDC-SR) that controls the number of dropped bundles by modifying MOFO. This is because MOFO may drop a large number of dropped bundles in order to accommodate new bundles whereas, due to the mobility node, the dropped bundles may be forwarded again to the same node in the future. This results in the network with high network overhead. In MDC-SR, they define an upper bound threshold for the number of buffered bundles such that if a node holds more buffered message count than an upper bound, the drop procedure will not be called.

In another work, Burns et al. [28] propose Meets and Visits (MV), a scheme that learns the frequency of meetings between nodes and how often they visit a certain region. This information is used to rank each bundle according to the likelihood of delivering a bundle through a specific path. However, many bundles with the same destination may exist in a node's buffer. Hence, in this case, all of them have the same priority to be forwarded whereas their different TTL values can affect bundle delivery. In another work, Pan et al. [27] propose a comprehensive buffer management policy based on state information such as node ID, list of buffered bundles and the five nodes that have the highest encounter rate. During routing, for a given bundle, a sender determines whether encountered nodes have recently met the bundle's destination. If so, the sender forwards the bundle to these nodes. It then arranges bundles in ascending order based on the

hop-count and number of forwards. Bundles with a hop-count greater than a threshold as well as having a size that is larger or equal to the size of a newly received bundle are selected for dropping and are arranged in ascending order based on the number of forwards. Accordingly, a node drops the bundle that has been forwarded the most. In another drop policy, Ayub and Rashid [26] propose T-drop, a policy that considers the size of bundles during congestion. Specifically, by defining a threshold range, a bundle is dropped if its size is within said threshold.

In [29], Fathima and Wahidabanu classify bundles based on their degree of importance into three priority queues: high, medium and low. When a node's buffer is full, those with a low priority are dropped first followed by those with a medium priority. Apart from that, they also consider the TTL value of bundles and specify that nodes do not drop their own bundles. In a similar work, Rohner et al. [30] propose an ordering policy that uses a relevance score to determine whether there is a match between a node's interests and a bundle's metadata.

Recently, Rashid et al. [24] proposed a drop policy called WBD that assigns a weight to each buffered bundle based on the message's properties such as size of bundle, its remaining lifetime, the residence time in the buffer, its hop-count and the replication's count. According to the weight of bundles, they are prioritized and the bundle with larger weight is dropped first. As mentioned in their paper, the replication's count is the number of relays that carry a given bundle and its value is obtained from time to live value of the bundle. However, they have not mentioned the correlation of bundle's lifetime and the number of relays.

In the schemes discussed thus far, references [15, 38] have considered classical drop/forward policies to deal with a limited bandwidth (short contact duration) and a finite buffer (congestion). However, these policies have not considered the parameters that are relevant to bundle delivery such as number of replicas. We will show in Section 4 that this can affect delivery ratios. Although references [17, 28] have considered using the number of replicas disseminated by a given node, this parameter does not represent the total number of disseminated replicas globally. In other words, for a given bundle, each node only knows the number of replicas that it has forwarded. In [25] and [27], the authors take advantage of encounter rates to estimate the probability of delivery. However, similar to references [17, 28], they do not know how many replicas have been disseminated throughout a DTN. None of the local knowledge schemes proposed thus far consider the number of disseminated replicas and/or number of replicas that will be disseminated in the future. This information can be

used to evaluate bundle delivery probability. However, under flooding-based protocols, it is impractical to obtain this information in order to make better decision when forwarding/discarding bundles. In QM-EBRP, we obtain this extra information locally, specifically information already maintained by quota-based protocols. In the next section, we will review global knowledge schemes and outline how they use the number of disseminated replicas and the number of nodes that have seen a given bundle.

2.2 Global knowledge schemes

RAPID [22] is the first protocol that considers both buffer and bandwidth constraints. RAPID assigns a utility to each bundle. A bundle's utility measures its expected contribution in maximizing a metric such as delay. RAPID replicates bundles that lead to the highest increase in utility. A key limitation of RAPID is that in order to derive bundle utilities, information about replicas has to be flooded throughout the network. This causes high overheads, and due to delays, the propagated information may be obsolete when it reaches nodes. Also, their results show that whenever traffic increases, their metadata channel consumes more bandwidth. This is undesirable because metadata amplifies the effects of congestion by occupying precious buffer space. In our work, we avoid these problems by using local information. In another work [23], Yong et al. present a drop policy that uses the control channel in [22] to help nodes obtain global network information such as transmission opportunities of bundles, node meeting times and duration. However, forwarding issue is not addressed. In [16], Kim et al. propose a drop policy to minimize the impact of buffer overflow. When buffer overflows, a node discards the bundle with the largest expected number of copies. This assumes keeping bundles with a small number of replicas increases delivery ratio.

Krifa et al. [20] introduce a distributed algorithm to approximate the number of replicas and number of nodes (excluding source) that have seen a bundle i since its creation. This estimation is based on the number of buffered bundles that were created before bundle i . As a result, this algorithm is dependent on the dissemination rate of previous bundles. This means any change in topology will result in inaccurate/obsolete information, especially for newly generated bundles [19]. In a similar work to [20], Yin et al. [31] propose an Optimal Buffer Management (OBM) policy to optimize the sequence of bundles for forwarding/discarding. They use a multi-objective utility function that considers metrics such as delivery, delay and overhead concurrently. In another work, Pan et al. [32] combine two routing protocols:

PROPHET [18] and binary Spray and Wait [43]. An encountered node is selected for forwarding based on PROPHET, and a number of replicas for forwarding is based on binary Spray and Wait. They use the bundle utility in [20] to drop bundles with the lowest utility value when a node's buffer is full. Moreover, if the last copy of a bundle is left at a sender and its utility is greater than a threshold, the last copy is forwarded. Otherwise the copy will remain at the sender. However, similar to [20], this method suffers from obsolete/inaccurate information.

Yun et al. in [49] propose a drop policy called the Average Forwarding Number based on Epidemic Routing (AFNER). In AFNER, when a node needs to receive an incoming message and its buffer is full, the node drops a bundle whose number of forwarded replicas is larger than the average number of the whole forwarded replicas in the network.

In a recent work [21], Krifa et al. propose a drop and forward policy that permits nodes to gather global knowledge at different times. Hence, during contacts, nodes flood information such as "a list of encountered nodes" and "the state of each bundle carried by them" as a function of time. However, due to large delays, this information may take a long time to propagate. The authors estimate the dissemination rate of a bundle based on the average dissemination rate of older bundles. However, the computed rate may have a large variance, causing errors when computing the resulting utility function. Elwhishi et al. [19] use the Markov chain model of [50] to predict the delay and delivery ratio under epidemic forwarding. However, as computing the stationary probabilities of the Markov chain incurs high computational complexity, they propose a forward/drop policy called Global History-based Prediction (GHP) that uses ordinary differential equations (ODEs). The ODEs, which calculate the utility of each bundle, incorporate two global parameters: the number of bundle copies and the number of nodes that have seen a bundle.

In [33], Liu et al. use a utility that estimates the total number of replicas and the dissemination speed of a bundle. Nodes update this information when they meet each other. During congestion and forwarding, a bundle that has the maximum utility value is dropped first and a bundle with the minimum utility value is forwarded first. Also, during forwarding, if the maximum utility of bundles in a sender's queue is smaller than the minimum utility value of bundles in a receiver's node, the sender forwards all its bundles to the receiver. Moreover, if the minimum utility value of bundles in a sender's queue is greater than the maximum utility value of bundles in a receiver's node, the sender will only forward bundles if the receiver has free space. In a similar work to [33], Shin and Kim [34] propose a

forward/drop policy that uses, (i) for a given bundle, an estimate of the total number of replicas, in a DTN, and (ii) for a given node, the number of replicas of a bundle it has replicated. Based on the said parameters and the elapsed time since a bundle was generated, a per bundle delivery utility is calculated. Also, a per bundle delay utility is derived from parameters (i) and (ii) and the bundle's remaining lifetime.

Ramanathan et al. [35] propose the PRioritized EPidemic scheme (PREP), a drop and forward policy for epidemic routing protocols. PREP prioritizes bundles based on source-destination cost and bundle expiry time. Here, cost is the average outage time of links on a path, and this information is flooded throughout a DTN and is used by the Dijkstra algorithm to compute the minimum source-destination cost. In their drop policy, a node with a full buffer first selects bundles that have a hop-count value greater than a threshold. Accordingly, selected bundles are sorted based on their cost to their intended destination and the bundle with the maximum cost is dropped first. In terms of transmission priority, if a bundle incurs a lower cost of delivery through an encountered node, the bundle with the longest remaining lifetime will be forwarded first. The main limitation of PREP is that it requires link cost to be flooded. However, due to large delays and dynamic topology, the computed path cost may become dated quickly.

Although the aforementioned policies are used to handle forward and drop bundles when congestion occurs, some other policies are also proposed to regulate congestion in the network [51]. For example, in [52], Coe and Rachavendra propose a token-based congestion control regulates the amount of traffic in the network based on network capacity. Network capacity is measured by the amount of data to be delivered within a given time period. The proposed scheme in [53] responds to congestion by limiting the number of bundles' replicas based on the current level of congestion in the network. The congestion level is an estimation of traffic amount at nodes that is collected during node encounters. Similarly, in [54], nodes broadcast their buffer occupancy to their neighbours. Then, this information is used to decide which nodes to forward. In [55], nodes use a migration algorithm to transfer bundle to nodes at which less congestion occurs.

2.3 Discussion

Table 1 shows a comparison of prior works and also to QM-EBRP. In summary, we make the following contributions with respect to the problem outlined in Section 1. First, the aforementioned local and global policies [15–17, 19–23, 25–35] are designed for flooding

protocols, e.g. [18, 42]. This means they are allowed to replicate a bundle without any limit. However, under quota-based protocols, if a replica is dropped, the bundle will have one fewer copy. This may reduce the probability of delivery. Although many schemes, e.g. [16, 17, 19–23, 27, 31–34], have considered the number of disseminated replicas to estimate the delivery probability, they do not take into consideration the remaining number of replicas that nodes are permitted to replicate. In contrast, our proposed policy, QM-EBRP, works under quota protocols, meaning we take into account the number of existing replicas and the remaining replicas to be disseminated in the future.

Second, in a DTN using a flooding protocol, buffer management is exacerbated by the difficulty in obtaining global knowledge of bundles and other nodes. The key questions to be answered include (i) how many replicas are distributed in a DTN, (ii) how many replicas of a bundle will be disseminated in the future and (iii) which bundles have already been delivered to their destination. Prior works [16, 17, 19–23, 27, 31–34] consider a bundle with a larger number of disseminated replicas to have a higher chance to be delivered. However, due to large delays, collected information may become obsolete. References [19, 21] address this problem by approximating the required information via a Gaussian distribution. However, the resulting estimates are not accurate under different forwarding strategies. To address this issue, we utilize three bundle properties available locally at each node: number of available replicas, maximum number of forwarded replicas and time to live. As we show in Section 3, these properties enable us to derive functions which calculate expected delay and the probability that a bundle has been delivered or will be delivered in the future. Then, we calculate the gradient of the said functions with respect to the number of available replicas and time to live in order to consider the rate of change in delivery probability when the parameters change. In turn, these rates of change enable the system to know how quickly the system reaches to maximum delivery probability. Accordingly, nodes prioritize the dropping and forwarding of bundles during congestion and at each contact.

3 System description

Let us consider a DTN where source nodes generate bundles periodically. Each bundle specifies the number of copies which a relay is allowed to create. Each bundle must be delivered to its destination within a given TTL. Moreover, each node records its rate of encounters with other nodes. This will be used to determine the forwarding priority of a bundle at each contact

and which bundles to drop when buffer overflows. We first describe system settings. Specifically, we first expound the routing protocol (forwarding strategy), mobility model and assumptions before formulating the problem precisely.

3.1 Routing

As mentioned, in this paper, we consider encounter-based quota protocols [3, 5], specifically EBR [5]. In details, EBR generates a finite number of replicas for each bundle. Every node running EBR is responsible for maintaining its past average rate of encounter with other nodes, which is then used to predict future encounter rates. To track a node's rate of encounter, the node maintains two pieces of local information: an encounter value (EV), and a current window counter (CWC).

The variable EV represents a node's past rate of encounters as an exponentially weighted moving average, whilst CWC is the number of encounters in the current time interval. EV is updated periodically to account for the most recent CWC. Specifically, EV is computed as follows:

$$EV = \alpha \times CWC + (1-\alpha) \times EV_{(old)} \quad (1)$$

where $\alpha \in (0, 1)$ is a weighting coefficient, i.e. $\alpha = 0.85$. In EBR, every 30 s, nodes' encounter rate is updated and the CWC is reset to zero. Equation (1) is inspired by studies [56, 57] on the characteristics of human mobility from real-world traces in where people usually roam in relatively small regions. This implies that maintaining a node's past average rate of encounters can be efficiently used in prediction of future encounter rate. In words, Eq. (1) is used to gradually adapt a node's encounter rate when the node is located in high- or low-node-density areas. Accordingly, to detect large shifts quickly, α is assumed to be a large value, e.g. $\alpha = 0.85$. The value of this parameter is clearly justified in EBR.

The primary purpose of tracking the rate of encounter is to decide how many replicas of a bundle a node will transfer during a contact opportunity. Hence, when nodes a and b meet each other, node a sends a proportional number of the i th bundle M_i based on the encounter rate of both sender and receiver. Specifically,

$$k = m_i \times \frac{EV_b}{EV_b + EV_a} \quad (2)$$

where m_i is the available number of replicas for the i th bundle at node a . The terms EV_a and EV_b respectively represent the encounter rate for nodes a and b . As a result, k replicas of bundle M_i are forwarded to node b .

We adopt EBR because of the following reasons. Firstly, it uses encounter rates when forwarding bundles.

In DTNs, nodes will naturally have varying rates of encounters [4, 5]. This parameter is used to derive the service rate of a node, which has a non-negligible impact on delivery ratio and delay. Secondly, EBR limits the number of replicas for each generated bundle. Therefore, for each bundle, a fixed number of replicas exist in the network that gives knowledge to each node to know the maximum number of replicas of each bundle that can be disseminated in the network. We emphasize that the routing process and scheduling policy are completely decoupled. This means that the next hop selection is decided based on a routing protocol, e.g. EBR, where a series of bundles are selected for forwarding whereas when a node's buffer is full and/or contact's duration is not sufficient to forward all selected bundles, a decision is made by a scheduling policy, i.e. QM-EBRP. This is the focus of the present work, and as we show in Section 5, this accounts for better performance.

3.2 Mobility model

Nodes change their location, velocity and acceleration over time. These parameters are governed by the mobility model. In general, mobility models [58–61] can be categorized into (i) map and (ii) random. Map-based models dictate the movement of nodes according to predefined paths and routes derived from real map data. In random mobility models, nodes do not follow any predetermined paths. However, random mobility models are not realistic as humans do not move randomly. Hence, in this paper, we consider mobility models, e.g. [58–62], where meeting time distribution has the following property of the mobility model assumed in [50, 58–63]. We assume the meeting times between nodes are distributed exponentially. As an example of such assumption, Karagiannis et al. in [62] used six distinct traces, namely UCSD [64], Vehicular [65], MITcell [66, 67], MITbt [66, 67], Cambridge [68, 69] and Infocom [68, 70] in order to demonstrate the exponential distribution of meeting times across all data sets. Here, “meeting” refers to the time when two nodes come within radio range of each other.

We make the following assumptions:

1. Each bundle has a finite number of replicas.
2. In order to replicate a bundle, a node will keep one replica for itself and the other replicas are forwarded to other nodes.
3. Each node has a finite buffer.
4. Short contact duration, meaning nodes do not have a sufficient bandwidth to empty their buffer.
5. Nodes have different speeds.
6. Nodes move independently of each other.

7. Mobility is heterogeneous, meaning different node pairs have different meeting rates.

4 Proposed queue management policy

Let us consider a contact between nodes i and j , with both nodes having limited resources, i.e. low data rate and buffer space. In this setting, there are two sub-problems:

- *Forward scheduling policy.* If node i has bundles to forward to node j , but is faced with a short contact duration or low data rate, both of which prevent it from forwarding all bundles to node j , the question then is to determine which bundles to forward such that the delivery ratio is maximized and the delay is minimized.
- *Bundle drop policy.* Consider when one or more bundles arrive at node j with a full buffer. The question then is to determine which bundles to discard whilst maximizing delivery ratio and minimizing delay.

The objective of both policies is to control congestion in order to improve delivery and delay. However, this becomes challenging when there are only a finite number of replicas, as is the case with quota protocols. To this end, we propose a Queue Management in Encounter-Based Routing Protocol (QM-EBRP) policy, designed specifically for quota-based protocols with the aim of maximizing (i) the delivery ratio of all bundles and (ii) the expected average delay of all delivered bundles.

4.1 Overview

Algorithm 1 presents the steps performed by QM-EBRP. Figure 2 provides an overview of QM-EBRP's functional modules and their relationships. Our algorithm starts whenever a connection is up (line 2). Upon contact, a node can either be in the send or receive mode, depending on the summary vector exchange during contact. In the receiving mode, for every bundle i in a receiver's buffer, the multi-objective utility $UF_i()$ is called to determine the bundle's utility. After that, bundles are sorted in ascending order. Finally, based on the sorted bundle list, *dropQueue* bundles are dropped from the head of the queue (lines 4–9). In the sending mode, the EBR [5] routing protocol selects bundles to forward. Hence, we will have a list of bundles for forwarding, called *forwardSelection*. In the next step, a multi-objective utility is calculated for every bundle in the *forwardSelection* list. Bundles are then sorted in descending order. Finally, bundles are dropped from the head of the sorted list *forwardQueue* (line 18).

Algorithm 1 QM-EBRP drop/forward policy

```

1- Start

2- while connection is up

3- if mode = receiving

4-  $ReceiverQueue \leftarrow$  bundles in receiver's buffer

5- for every bundle  $i$  in  $ReceiverQueue$ 

6-  $UF_i \leftarrow$  multi_objective_utility_function( $i$ )

6- end for

8-  $dropQueue \leftarrow$  sort( $UF$ ,  $ReceiverQueue$ , 'increase')

9- DROP( $dropQueue$ )

10- end if

11- if mode = sending

12-  $SenderQueue \leftarrow$  bundles in sender's buffer

13-  $forwardSelection \leftarrow$  EBR( $SenderQueue$ )

14- for every bundle  $i$  in  $forwardSelection$ 

15-  $UF_i \leftarrow$  multi_objective_utility_function( $i$ )

16- end for

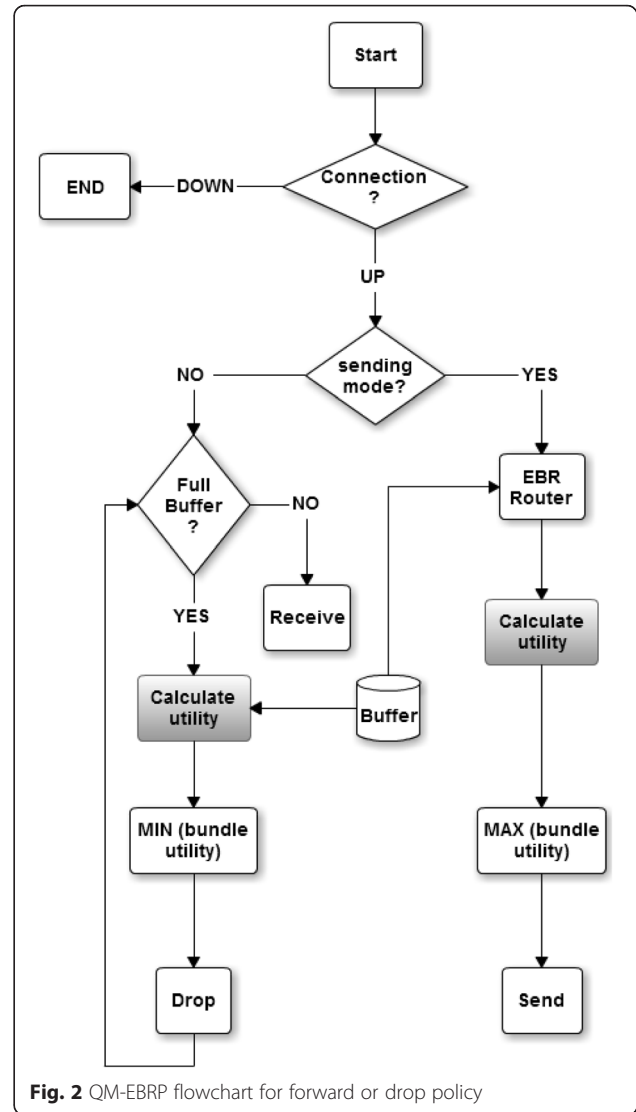
17-  $forwardQueue \leftarrow$  sort( $UF$ ,  $forwardSelection$ , 'decrease')

18- FORWARD( $forwardQueue$ )

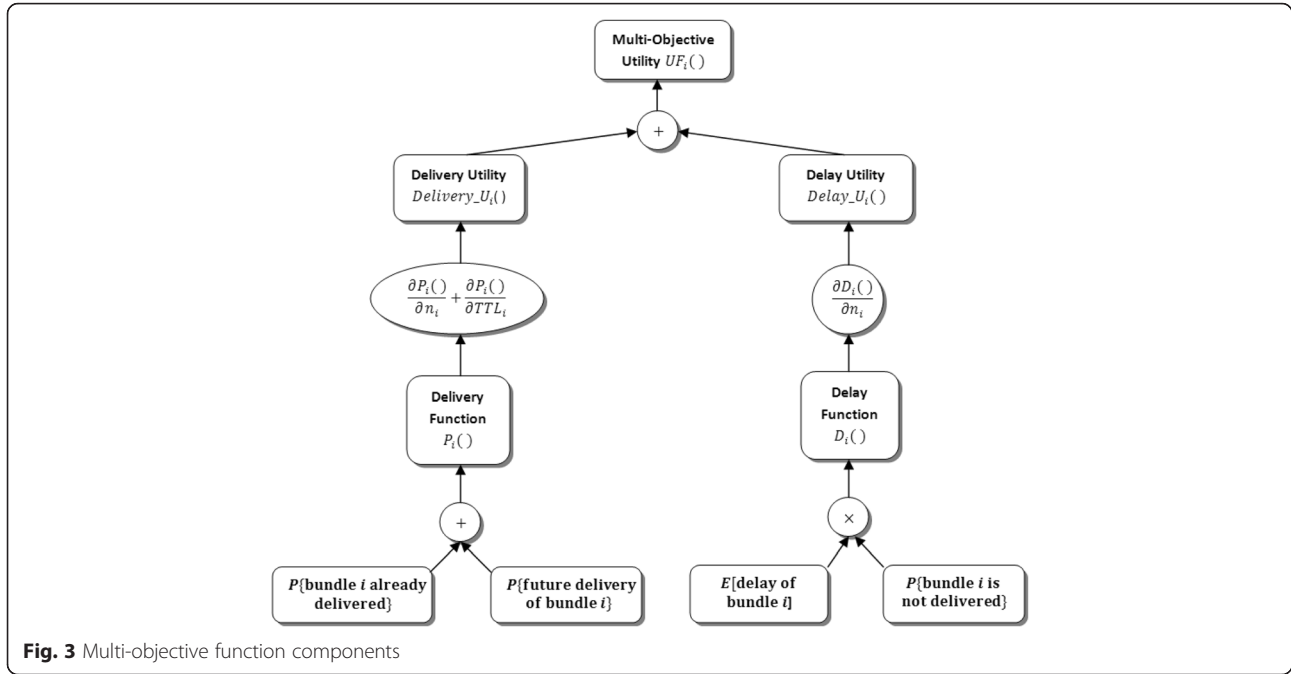
19- end if

20- End

```



A key module used by QM-EBRP is the multi-objective utility function, which uses the delay and delivery function. Figure 3 depicts the components of the proposed multi-objective function. Briefly, as we will explain in Section 4.2, the delivery function $P_i()$ considers the probability of delivery for every bundle i . To calculate the delivery probability, we need to calculate how likely bundle i has been delivered or will be delivered in the future. This is carried out, for a given bundle i , using the number of disseminated replicas and the number of replicas that will be disseminated in the future. The delay function considers the expected delay E_i of bundle i , if the bundle is not yet delivered (details in Section 4.3). The expected delay of bundle i is the time until the first copy of bundle i is delivered to its destination. Given both functions, we use their rate of change with respect to two parameters, namely, the number of current replicas (n_i) and bundle's lifetime



(TTL_i) to derive a bundle i 's maximum delivery ratio and minimum delay; see Sections 4.2.1 and 4.3.1. Both functions are then used in a multi-objective function, which is then responsible for prioritizing bundles during congestion and forwarding. Table 2 lists a summary of all notations used in the following sections.

4.2 Delivery function

Let L denote the number of nodes. We denote the number of bundles at time t by $K(t)$. Each bundle has N replicas. Assume that each node j has a meeting rate $M_j(t)$ and each bundle i has a lifetime at time t of $TTL_i(t)$. In fact, $M_j(t)$ is obtained from EBR that determines the node j 's encounter rate. Recall that the value of $M_j(t)$ for each node is different over time. Hence, the probability that a copy of bundle i will not be delivered by node j is dependent on the probability that node j 's next meeting time with

the destination is greater than $TTL_i(t)$. This probability is equal to $\exp(-M_j(t) \times TTL_i(t))$.

For each bundle $i \in [1, K(t)]$, let $n_i(t)$ be the number of replicas of bundle i that a node has in its buffer at time t . Also, denote $m_i(t)$ the number of replicas of bundle i that has been forwarded to other nodes up to time t , i.e. we have $n_i(t) + m_i(t) = N$. For example, a source node generates bundle i with 10 replicas ($N = 10$); after two contacts with other nodes, only three replicas are left at source node ($n_i(t) = 3$). Hence, the maximum number of replicas that has been disseminated throughout the network is seven ($m_i(t) = 7$). We also define "A" and "B" to be the event "bundle i has not been delivered" and "bundle i will not be delivered in the future", respectively. Then, if we know bundle i has $n_i(t)$ available replicas at node j at time t , we have the following conditional probability:

Table 2 Summary of notations

Variable	Description
L	Number of nodes
$n_i(t)$	Number of available replicas of bundle i at a node at time t
$TTL_i(t)$	Remaining time to live for bundle i
$m_i(t)$	Number of forwarded replicas of bundle i up to time t
$N_i = n_i(t) + m_i(t)$	Total number of replicas for bundle i
$M_j(t)$	Node j 's encounter rate at time t
$K(t)$	Number of bundles in the system at time t
T_i	Elapsed time for bundle i

$$\begin{aligned}
 P_i\{B|A\} &= \prod_{j=1}^{n_i(t)} \exp(-M_j(t) \times TTL_i(t)) \\
 &= \exp(-M_j(t) \times n_i(t) \times TTL_i(t))
 \end{aligned} \quad (3)$$

Equation (3) considers how likely node j will not deliver bundle i with $n_i(t)$ available replicas. Note that this equation does not take into account whether a copy of bundle i has been delivered up to time t . Hence, if we assume all nodes including bundle i 's destination have the same chance to receive bundle i , the probability that one of the $m_i(t)$ replicas of bundle i has been delivered is

$$P_i\{\bar{A}\} = \frac{m_i(t)}{L-1} \quad (4)$$

where \bar{A} corresponds to the event “bundle i is delivered”. Notice that the system ensures that the number of forwarded replicas throughout the network is not greater than $m_i(t)$. Hence, in this calculation, we know that the probability that a bundle is delivered is not greater than a threshold. Combining Eqs. (3) and (4), the probability that a bundle i with N replicas will be delivered before its TTL expires is

$$\begin{aligned} P_i &= P_i\{A\} \times (1 - P_i\{B|A\}) + P_i\{\bar{A}\} \\ &= \left(1 - \frac{m_i(t)}{L-1}\right) \times \left(1 - \exp(-M_j(t) \times \text{TTL}_i(t) \times n_i(t))\right) + \frac{m_i(t)}{L-1} \end{aligned} \quad (5)$$

In words, Eq. (5) calculates the delivery probability of each bundle. Hence, the global delivery ratio (DR) of all existing bundles at time t is calculated as follows:

$$\text{DR} = \sum_{i=1}^{K(t)} \left[\left(1 - \frac{m_i(t)}{L-1}\right) \times \left(1 - \exp(-M_j(t) \times \text{TTL}_i(t) \times n_i(t))\right) + \frac{m_i(t)}{L-1} \right] \quad (6)$$

4.2.1 Delivery utility

To maximize the delivery ratio, we calculate the rate of change with respect to $n_i(t)$ and $\text{TTL}_i(t)$. Specifically, the gradient of the delivery ratio is

$$\nabla P_i = \frac{\partial P_i}{\partial n_i(t)} dn_i(t) + \frac{\partial P_i}{\partial \text{TTL}_i(t)} d\text{TTL}_i(t) \quad (7)$$

where $\frac{\partial P_i}{\partial n_i(t)}$ and $\frac{\partial P_i}{\partial \text{TTL}_i(t)}$ are the rate of change of the delivery ratio with respect to $n_i(t)$ and $\text{TTL}_i(t)$ and are defined as follows:

$$\frac{\partial P_i}{\partial n_i(t)} = \left(1 - \frac{m_i(t)}{L-1}\right) \times M_j(t) \times \text{TTL}_i(t) \times \exp(-M_j(t) \times \text{TTL}_i(t) \times n_i(t)) \quad (8)$$

$$\frac{\partial P_i}{\partial \text{TTL}_i(t)} = \left(1 - \frac{m_i(t)}{L-1}\right) \times M_j(t) \times n_i(t) \times \exp(-M_j(t) \times \text{TTL}_i(t) \times n_i(t)) \quad (9)$$

The maximal directional derivative is then

$$\text{Delivery_}\mathcal{U}_i = \sqrt{\left(\frac{\partial P_i}{\partial n_i(t)}\right)^2 + \left(\frac{\partial P_i}{\partial \text{TTL}_i(t)}\right)^2} \quad (10)$$

As we will see later, QM-EBRP uses Eq. (10) as the delivery utility for a copy of bundle i with respect to the total delivery rate.

4.3 Delay function

We now consider delay. Let X_i be a random variable corresponding to the delay of bundle i . Also, let T_i be the elapsed time for bundle i . In other words, it measures the time since bundle i was generated by its source node. Then, the expected delay for bundle i for which none of its copies are delivered is given by

$$D_i = \left(1 - \frac{m_i(t)}{L-1}\right) \times E[X_i > T_i] \quad (11)$$

The mean or expected value of an exponential distribution with rate parameter λ is $\frac{1}{\lambda}$ [71]. As mentioned in [21], the time until the first copy of bundle i reaches the destination via node j follows an exponential distribution with rate parameter $M_j(t) \times n_i(t)$. Hence, the mean or expected value of this distribution is $\frac{1}{M_j(t) \times n_i(t)}$ [21]. It follows that

$$E[X_i > T_i] = T_i + \frac{1}{M_j(t) \times n_i(t)} \quad (12)$$

Substituting Eq. (12) into Eq. (11), we get

$$D_i = \left(1 - \frac{m_i(t)}{L-1}\right) \times \left(T_i + \frac{1}{M_j(t) \times n_i(t)}\right) \quad (13)$$

Hence, D_i is the expected delay for each bundle i . The following equation is used to calculate the average delay (AD) of all bundles at time t :

$$\text{AD} = \sum_{i=1}^{K(t)} \left[\left(1 - \frac{m_i(t)}{L-1}\right) \times \left(T_i + \frac{1}{M_j(t) \times n_i(t)}\right) \right] / K(t) \quad (14)$$

4.3.1 Delay utility

We next turn our attention to minimizing the average delay. Equation (15) represents the delay utility for bundle i . We derive the rate of change for delay, see Eq. (13), in the direction of the negative gradient with respect to $n_i(t)$. The derived equation represents how fast a bundle will be delivered. This means a bundle with a large delivery utility will experience minimum delay. Hence, a node needs to apply the following delay utility for each bundle i :

$$\text{Delay_}\mathcal{U}_i = \frac{\partial D_i}{\partial n_i(t)} = \left(-1 + \frac{m_i(t)}{L-1}\right) \times \left(\frac{1}{M_j(t) \times n_i(t)^2}\right) \quad (15)$$

4.4 Multi-objective utility function

We use a multi-objective function that incorporates delivery (see Eq. (10)) and delay utility (see Eq. (15)).

Briefly, a multi-objective utility function is represented as the following multi-objective optimization problem:

$$\min(f_1(x), f_2(x), \dots, f_k(x)) \quad \text{or} \quad \max(f_1(x), f_2(x), \dots, f_k(x)) \quad (16)$$

where the integer $k \geq 2$ is the number of objectives and x is a vector of decision variables in the set X . A key issue when incorporating the said utilities is that their values are in a different domain. For example, the domain of the delivery utility belongs to \mathbb{R}^+ and for the delay utility, it is \mathbb{R}^- . To this end, we normalize the delay and delivery utility as follows:

$$\varphi(\text{Delivery_}U_i) = \frac{\text{Delivery_}U_i - \mu_{\text{dvu}}}{\sigma_{\text{dvu}}} \quad (17)$$

where μ_{dvu} is the mean of delivery utility of all bundles in a node's queue. Also, σ_{dvu} is the standard deviation of delivery utility of the considered bundles. The same procedure applies to Delay_ U_i . Specifically,

$$\varphi(\text{Delay_}U_i) = \frac{\text{Delay_}U_i - \mu_{\text{dlu}}}{\sigma_{\text{dlu}}} \quad (18)$$

where μ_{dlu} is the mean of delay utility of all bundles in a node's queue. Also, σ_{dlu} is the standard deviation of delay utility of the considered bundles. Hence, the multi-objective utility function U_i used by QM-EBRP is as follows:

$$U_i = \alpha \times \varphi(\text{Delivery_}U_i) + \beta \times \varphi(\text{Delay_}U_i) \quad (19)$$

The coefficients α and β determine the impact of delivery and delay on the multi-objective utility function, respectively. In this paper, we investigate delivery and delay equally, meaning $\alpha = \beta = 1$. In words, Eq. (19) represents how fast bundle i reaches the maximum delivery rate and minimum delay. Hence, if bundle i has a greater utility value than bundle j , bundle i will have a higher delivery probability and lower delay. Hence, in this paper, we use Eq. (19) in order to obtain the utility for each bundle.

5 Evaluation

Our experiments are conducted in the Java-based simulator, Opportunistic Network Environment (ONE) [72]. It is able to generate node movements using different mobility models. Example mobility models [58–60] include the shortest map-based model, working day movement model and random walk model.

We evaluate QM-EBRP against six local knowledge policies and one optimal global knowledge policy. We first present a brief description of the following local knowledge policies: DO, LIFO, FIFO, MOFO, LEPR and drop greatest HOP-COUNT. We briefly describe how

each said policy is used as a drop and forward policy. DO drops the oldest bundle if a node's buffer is full and forwards the bundle that has the maximum lifetime. LIFO drops the last arriving bundle and forwards the bundle at the head of the queue. FIFO drops the bundle at the head of the queue and forwards the last bundle that has arrived. In MOFO, every node maintains a variable FP, which is initialized to zero, for each bundle. Each time a bundle is forwarded, FP is updated according to Eq. (20), where P is the delivery probability that is used in PROPHET [18].

$$FP = FP_{\text{old}} + P \quad (20)$$

The bundle that has been forwarded the most, i.e. the highest FP, is dropped first, and the bundle that has been forwarded the least, i.e. the lowest FP, is forwarded first. LEPR drops the bundle with the lowest delivery probability. In other words, LEPR drops the bundle that has the lowest P . Lastly, HOP-COUNT drops the bundle that has the greatest number of hops and forwards the bundle that has the smallest number of hops. We also evaluated QM-EBRP against Optimal Global Knowledge (OGK), a scheme that is similar to [21] and [33]. In this policy, we assume that nodes are synchronized with a shared global memory to update bundle information such as the number of disseminated replicas. Accordingly, every node is instantly aware of the accurate number of disseminated replicas of each bundle in the network. This policy thus allows us to compare QM-EBRP against a theoretical scheme.

We categorize our experiments into three groups based on mobility models. Specifically, we use ONE's default setting, whereby in the first group of experiments, the shortest map-based model is considered in a 5×3 km² area of downtown Helsinki, Finland. There are 60 nodes, each with a radio range of 20 m. We first assume all nodes have infinite buffer space and the speed of nodes is varied from 0.5 to 60 m/s, at an increment of 10. This causes nodes to have different contact durations. After that, we assume all nodes have finite buffer space and move at a constant speed of 30 m/s. We vary nodes' buffer space from 5 to 40 bundles, where the buffer size is double that of the previous experiment, i.e. 5, 10, 20 and 40 bundles. Lastly, we study the scenario where nodes have space for five bundles and the number of source/destination is varied from 10 to 60. In this experiment, bundles have a 60-min lifetime and the simulations last for 12 simulated hours and each data point is an average of 20 runs.

In the second experiment group, the working day movement of 60 people and 50 taxicabs is simulated in a 10×8 km² area of Manhattan, NY, USA [72]. People use their car with a probability of 0.5 to go shopping or work. Otherwise, they have to walk or catch a taxicab

with a probability of 0.5. Cars and taxicabs move at a minimum speed of 20 m/s and a maximum speed of 30 m/s, and pedestrians move at 2 m/s. Note that nodes are either at home, working or carrying out other activities such as shopping and meetings. These activities are deemed to be the most common and capture a typical working day for most people [73]. This experiment evaluates the network performance when the buffer space is varied from 10 to 70 bundles in increments of 10 bundles. All nodes are equipped with a radio range of 30 m.

In this experiment, bundles have an 8-h lifetime and the simulations last for three simulated days.

In the third group of experiments, 60 nodes with a radio range of 30 m move randomly in a $2 \times 2 \text{ km}^2$ area. This experiment evaluates the network performance when the buffer space is varied from 10 to 200 bundles in increments of 20 bundles. Bundles have a 5-h lifetime, and the simulations last for 24 simulated hours. Note that, in all experiments, the bundle size is 100 KB, and sources generate a bundle every 10 s. All nodes, upon

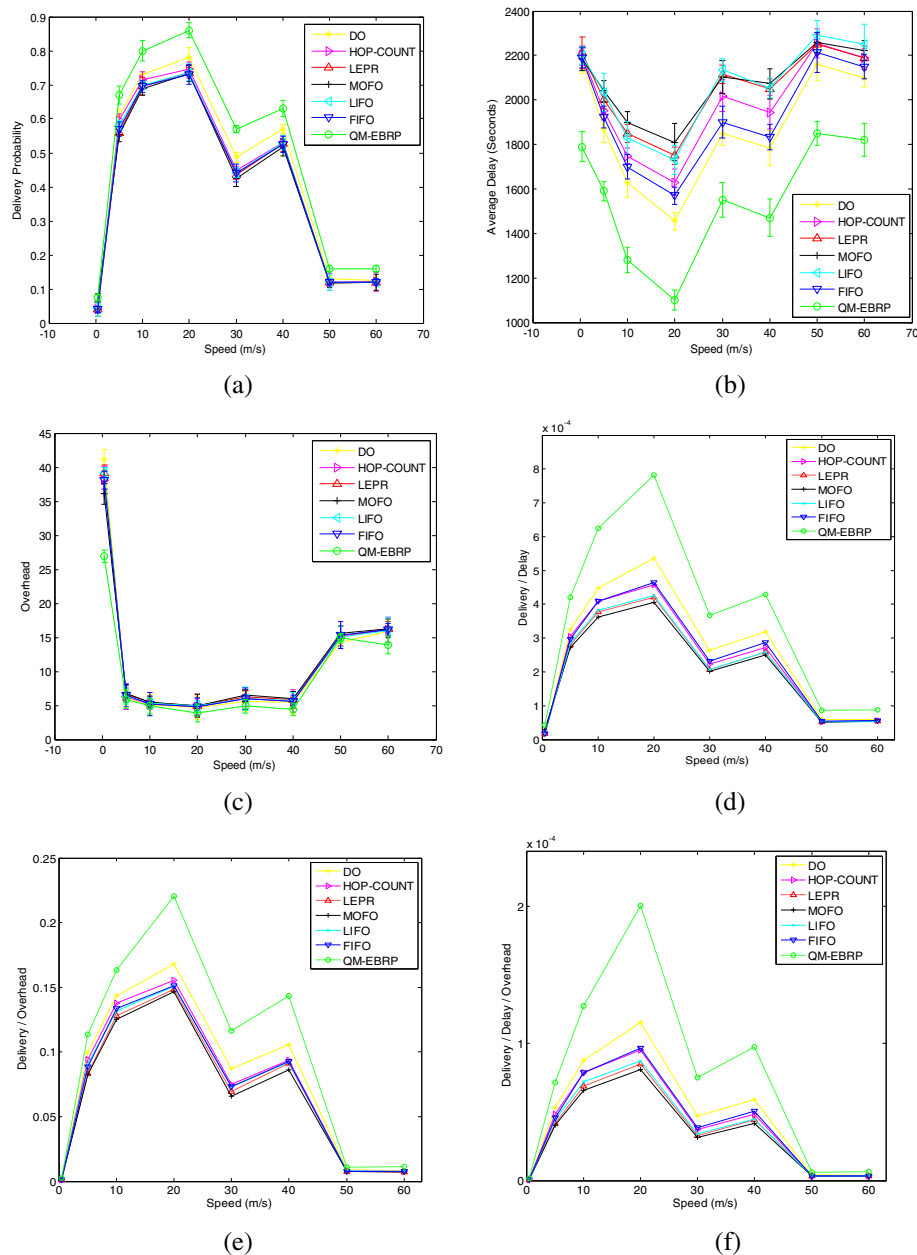


Fig. 4 Network performance under the shortest map-based mobility with different node speeds. **a** Delivery probability. **b** Average delay. **c** Overhead. **d** DA. **e** DOR. **f** DAO

contact, have a transmission speed of 100 Kbps. Also, each data point is an average of 10 runs, with minimum and maximum confidence intervals.

We consider three conventional performance metrics as well as introducing three other metrics used by the authors of EBR [5] to show the relative relationship between conventional metrics. Conventional metrics used include (1) *delivery probability*, defined as the ratio between the number of delivered bundles to the number of generated bundles; (2) *overhead*, defined as the ratio of the number of delivered bundles and number of carrier nodes; and (3) *average delay*, defined as the time from when a bundle is generated to its reception time. Whilst these three conventional metrics provide a comprehensive comparison, many protocols optimize one metric at the expense of another. Consider a protocol that delivers bundles quickly by preferentially using routes with a small number of hops. Otherwise, it does not forward bundles. Consequently, the protocol has a low overhead but delivery ratio is low. To overcome this issue, the following composite metrics are used to penalize protocols that unfairly optimize a metric. Briefly, Eq. (21) defines DA based on delivery ratio (DR) and average delay (AD).

$$DA = DR \times \frac{1}{AD} \quad (21)$$

In other words, DA scales the performance accordingly if a protocol optimizes for delivery ratio but has poor delay. Equation (22) defines DOR based on DR and overhead ratio (OR), i.e.

$$DOR = DR \times \frac{1}{OR} \quad (22)$$

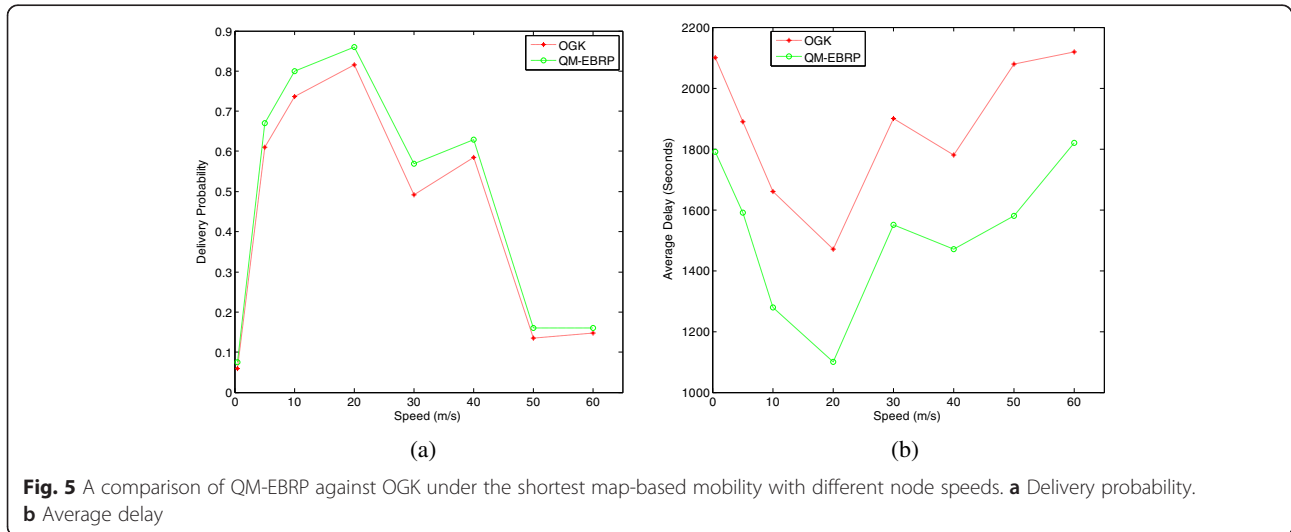
Hence, DOR captures the trade-off between DR and resulting overheads. Lastly, Eq. (23) defines DAO based on DR, AD and OR.

$$DAO = DR \times \frac{1}{AD} \times \frac{1}{OR} \quad (23)$$

In other words, DAO quantifies the performance of a protocol that myopically optimizes delivery ratio at the expense of average delays and overheads.

5.1 Shortest map-based mobility

Figure 4 shows the impact of speed and radio range when nodes have infinite buffer space. Hence, we do not consider drop policy. Recall that in the first scenario, nodes have different speeds, which help to simulate different contact duration. That is, when nodes' speed increases, contact periods become shorter and nodes cannot forward all queued bundles during contacts. In Fig. 4a, we find that the policies that do not use bundle information such as TTL result in low delivery ratios. For example, FIFO, HOP-COUNT, LEPR, MOFO and LIFO have a delivery ratio between 70.5 and 71.3 %. These policies prioritize bundles based on information such as arrival time, nodes' encounter rate and number of relays. Hence, for said policies, nodes may receive old bundles that do not have sufficient lifetime. Recall that the main reason for using bundle lifetime is to avoid forwarding old bundles during contact. For example, DO sends the bundle that has the longest remaining lifetime. We see DO has 5 % better delivery performance as compared to the said policies. Now, consider the scenario where node A has stored a bundle that has a large lifetime but the bundle has no more replicas to be forwarded. Accordingly, if node A meets the bundle's destination, the bundle will be delivered. Otherwise, it will never leave node A until its lifetime expires. In QM-EBRP, a higher forward priority is given to bundles that have a large lifetime and those that will generate a large number of replicas in the future. As shown in Fig. 4a,



QM-EBRP performs up to 15 % better than other policies in terms of bundle delivery. Note that, at speeds of 0.5 and 60 m/s, all the considered forward/drop policies have similar delivery probability. This is because at low speeds, nodes are within each other's range for sufficiently long, thereby allowing them to drain their queue. On the other hand, at high speeds, a contact may not be sufficient to transmit even one bundle. Consequently, delivery ratio reduces significantly. In terms of delay, as shown in Fig. 4b, policies that forward newly generated

bundles or recently transmitted bundles achieve a low delay. For example, DO, FIFO and HOP-COUNT have a delay of 1450, 1590 and 1630 s, respectively. QM-EBRP trades off delivery ratio and delay such that bundles' expected delay reduces and delivery ratio increases. Figure 4b shows that QM-EBRP delivers bundles up to 25 % quicker as compared to DO. We also found policies may deliver a small number of bundles quickly using a small number of hops. In this case, the overhead and delay reduces but the network experiences a low delivery ratio.

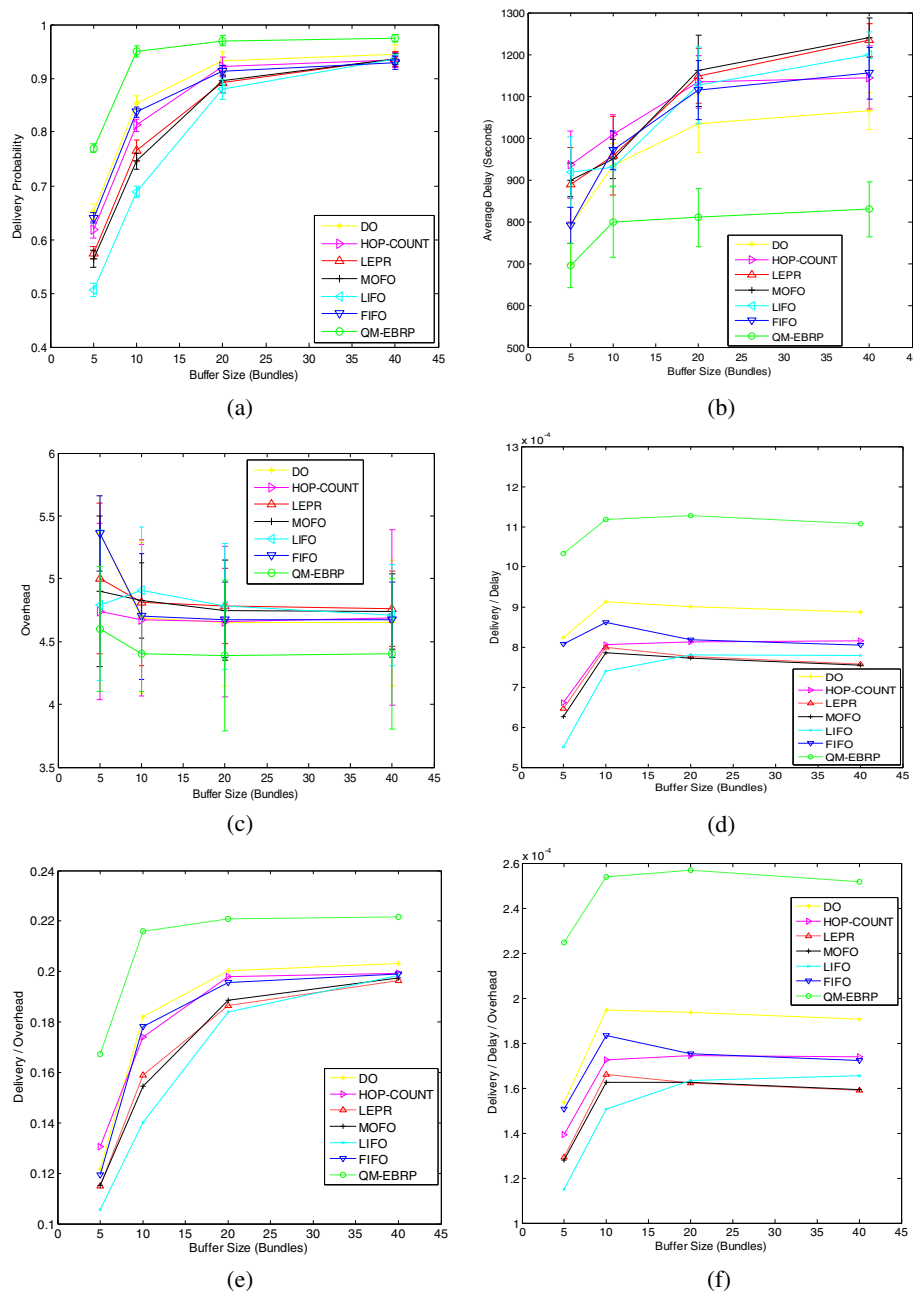


Fig. 6 Network performance under the shortest map-based mobility with different node buffer sizes. **a** Delivery probabilities. **b** Average delays. **c** Overheads. **d** DA. **e** DOR. **f** DAO

Figure 4d shows the trade-off between delivered bundles and delays. QM-EBRP recorded 60 % improvement in terms of DA. Figure 4e shows that QM-EBRP has up to 32 % improvement in terms of DOR. Also, Fig. 4f shows that QM-EBRP improves DOA up to 80 %.

Figure 5 shows a comparison of QM-EBRP against OGK. Although OGK does not suffer from inaccurate/obsolete information, it disregards information such as the lifetime of bundles and the encounter rates of nodes. This causes OGK to give a high priority to bundles that have a large number of replicas despite their short lifetime. The results in Fig. 5a show that QM-EBRP has 10 % more delivered bundles. Also, Fig. 5b shows that QM-EBRP has up to 25 % reduction in delay as compared to OGK.

In the next experiment, we consider different buffer sizes. We find that although increasing nodes' buffer size causes nodes to store more bundles, it can result in a high ratio of dropped bundles when long contacts occur. On the other hand, increasing nodes' buffer size causes nodes to select a larger number of bundles for forwarding over short contacts. QM-EBRP will lower the priority of a bundle with a lower delivery probability and larger delay. Note that a bundle has a low delivery probability if the dissemination rate is low and/or its remaining lifetime is short. Figure 6a shows that QM-EBRP has up to 12 % improvement in terms of delivery ratio as compared to DO. LIFO has the worse delivery ratio with 5 % fewer delivered bundles as compared to MOFO and LEPR. This is because LIFO drops recently received bundles. We found that delivery ratio gradually increases when nodes' buffer size increases. This is because nodes have the capability to buffer more bundles. In contrast, we see that in Fig. 6b, delivery delay also increases. This can be explained as follows. Suppose that contact duration is short. When nodes have a small buffer size, i.e. five bundles, nodes are able to drain their queue. On the

other hand, when nodes have a large buffer size, i.e. 20 and 40 bundles, they can only transmit a small portion of queued bundles. In this case, a large number of bundles may not be forwarded for a long time. This results in increased delay. In terms of delay, Fig. 6b shows that QM-EBRP has up to 16 % reduction as compared to DO and up to 23 % as compared to FIFO and HOP-COUNT. In terms of overheads, forwarding bundles that have a low delivery probability increases overhead. This is because forwarding these bundles increases the number of relays even though they may not have a chance to be delivered. QM-EBRP addresses this problem by giving a low priority to bundles that have a low delivery probability. From Fig. 6c, we see that QM-EBRP has up to 7 % reduction in overhead. To quantify the trade-off between delivery and delay, Fig. 6d depicts that QM-EBRP has up to 23 % improvement in DA. Also, Fig. 6e shows the trade-off between delivery and overhead that QM-EBRP has up to 22 % improvement in DOR. In terms of the trade-off between delivery, delay and overhead, Fig. 6f shows that QM-EBRP has up to 30 % improvement in terms of DAO.

Figure 7 compares QM-EBRP against OGK. In terms of delivery, Fig. 7a shows that QM-EBRP has up to 12 % improvement. As we mentioned earlier, when we increase the buffer size of nodes, a large number of bundles may not be forwarded for a long time. However, OGK does not consider the expected delay when forwarding bundles. Hence, bundles experience a large delay of 990 s. The performance of OGK versus QM-EBRP exhibits a similar trend for the forthcoming mobility models. We thus omit them from the paper.

Figure 8 shows the impact of different numbers of source/destination nodes. Suppose that only one destination exists in the northern part of a city and the source is in the southern part of the city. Hence, nodes forward

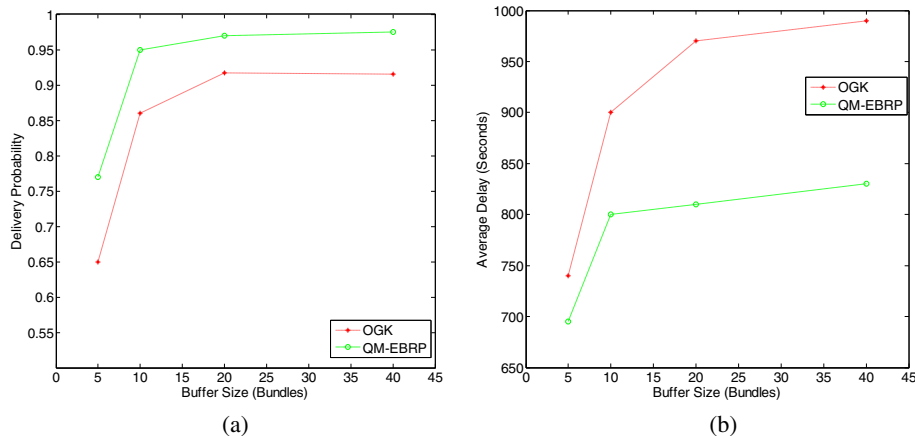


Fig. 7 A comparison of QM-EBRP and OGK under the shortest map-based mobility with different node speeds. **a** Delivery probabilities. **b** Average delays

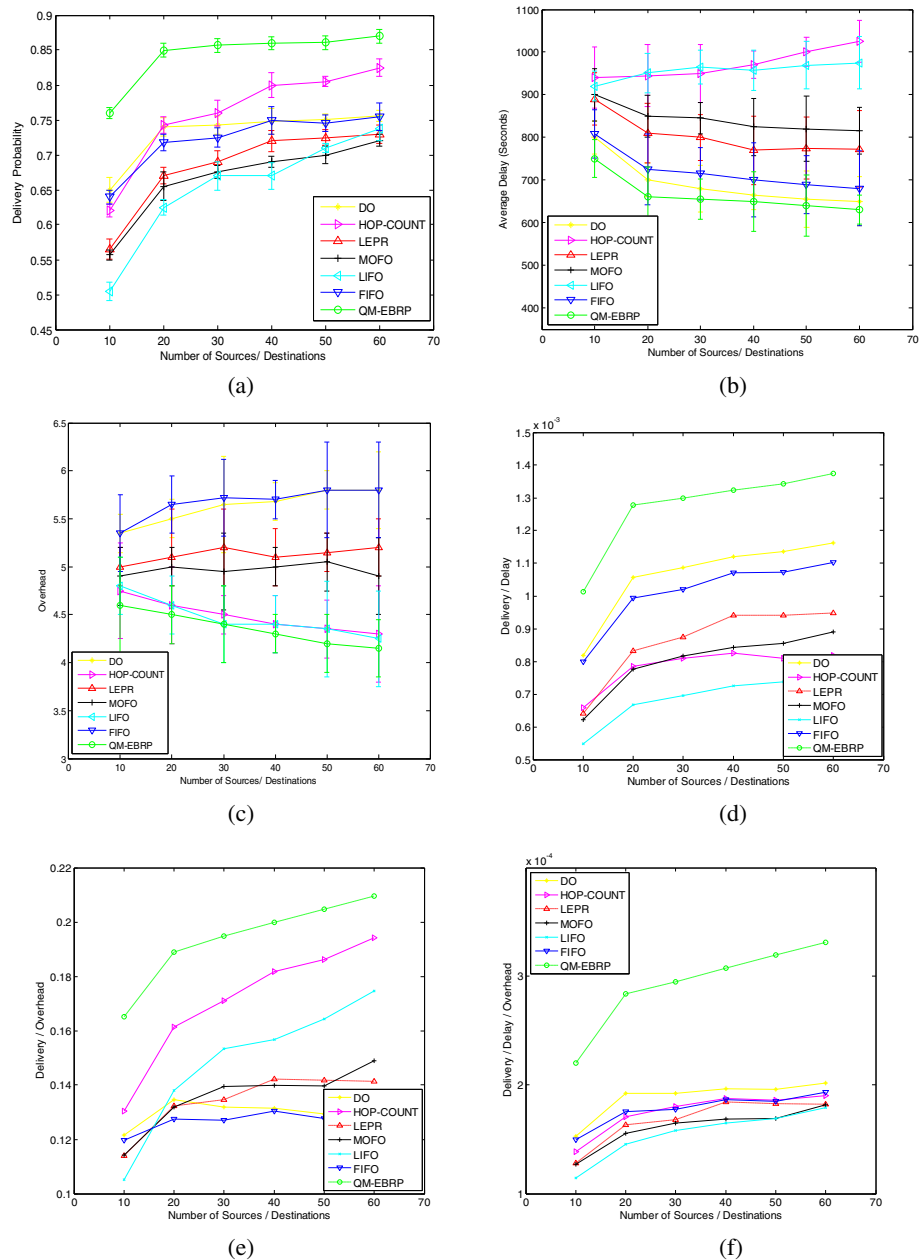


Fig. 8 Network performance under the shortest map-based mobility with different number of source/destination pairs. **a** Delivery probability. **b** Average delay. **c** Overhead. **d** DA. **e** DOR. **f** DAO

bundles towards the northern part of the city and consequently, nodes in that area experience a high load and thus drop bundles frequently. This example illustrates the downside of forwarding all bundles towards a small number of destinations, i.e. 10. Indeed, in our experiments, it results in protocols with low delivery ratios and large delays. For example, DO, FIFO and HOP-COUNT have a delivery ratio of 65, 64 and 62 %, respectively. Now, suppose there are multiple, geographically dispersed destination nodes. This means traffic will be distributed

uniformly across the network. Hence, when the number of destinations increases, the drop ratio of bundles decreases, resulting in a higher delivery ratio and smaller delays. Furthermore, destination nodes may not be reachable within a bundle's lifetime. To address the said issues, QM-EBRP takes advantage of nodes' encounter rate, bundle lifetime and number of bundle replicas to effectively consider how likely one of the bundle's replicas will be delivered within the bundle's lifetime. As shown in Fig. 8a, as compared to HOP-COUNT, DO

and FIFO, QM-EBRP has up to 17 % improvement in delivery ratio and also up to 7 % reduction in delay. In terms of DA, Fig. 8d shows that QM-EBRP has up to 24 % improvement. Also, Fig. 8f shows that QM-EBRP has up to 60 % improvement in terms of DAO.

We also compare our simulation results with the analytical results in order to show how highly they are correlated (see Fig. 9). Analytical results use Eq. (6) to calculate delivery probability. The 12-h network running is divided into 100 time units in which the total delivery probability is calculated based on the simulation and analytical model. As shown in Fig. 9, analytical results have higher delivery probability. This is because in the analytical model, facing with full buffer space is disregarded. To this end, despite a time shift, there is a high correlation between simulation and analytical results. The time shift is due to the analytical model calculates delivery probability for the bundles which have not been delivered whereas the simulation model calculates the delivery probability based on the number of delivered bundles and number of generated bundles. As a result, it takes time since the delivery of not delivered bundles are calculated until they will be delivered.

5.2 Working day movement model

Figure 10 depicts the network performance when nodes have different buffer sizes. We increase the simulation duration and bundles' TTL based on working hours to ensure every bundle has enough time to be delivered. We found that bundles' lifetime directly impacts delivery ratio. Accordingly, the policies that consider bundles' lifetime have a high delivery ratio. For example, DO delivers 70 % of bundles when nodes have a buffer size of 10 bundles. FIFO also indirectly considers bundle's TTL such that new arrival bundles are sent upon contact. The results in Fig. 10a show that FIFO delivers 69 % of the total bundles. Similar to Section 5.1, QM-EBRP takes advantage nodes' encounter rate. Figure 10a shows that QM-EBRP has up to 10 % improvement in

terms of delivery ratios. As for delays, we see in Fig. 10b that QM-EBRP recorded a 20 % drop. Figure 10c shows that QM-EBRP has 10 % less overheads. In terms of trade-off between delivered bundles and delays, Fig. 10d shows that QM-EBRP has up to 30 % improvement. In total, QM-EBRP achieves up to 35 % improvement.

5.3 Random mobility model

In another scenario, we consider the impact of different buffer sizes when nodes have random movement. A key issue in this model is the inability to predict nodes' future contacts via their encounter rates. Now, suppose that a large number of nodes are randomly located in an area and meet each other frequently for a short period of time. In this case, the nodes' encounter rate increases but nodes may not meet each other in the future as nodes do not follow any predetermined paths. Hence, nodes' encounter rate will be obsolete/inaccurate for future decisions. In this respect, QM-EBRP relies on other parameters such as the number of replicas and their TTL to prioritize bundles. Our simulation results in Fig. 11a show that in terms of delivery, QM-EBRP has up to 10 % improvement as compared to DO and up to 27 % improvement as compared to LEPR, HOP-COUNT, MOFO, LIFO and FIFO. In contrast, we see that MOFO has the lowest delivery ratio at 65 %. This is because MOFO considers delivery probability of bundles based on nodes' encounters, which is highly inaccurate in this mobility model. In terms of delay, Fig. 11b shows that QM-EBRP has a delay of 5050, 5400 and 5500 s when nodes' buffer size is 30, 90 and 200 bundles, respectively. We found that using nodes' encounter rate under a random mobility model causes inaccurate expected delay calculation. However, QM-EBRP also considers the number of disseminated replicas to estimate how likely a bundle will be delivered. Consequently, as compared to LIFO and LEPR, QM-EBRP has up to 16 % reduction in delay and up to 30 % reduction in delay as compared to MOFO. In terms of DAO, Fig 11f shows that

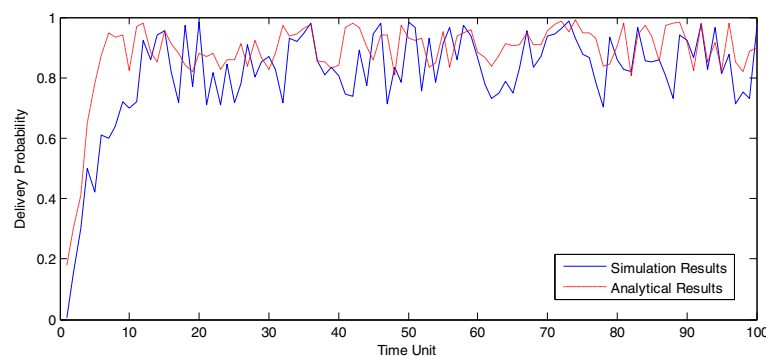


Fig. 9 Comparing simulation and analytical results

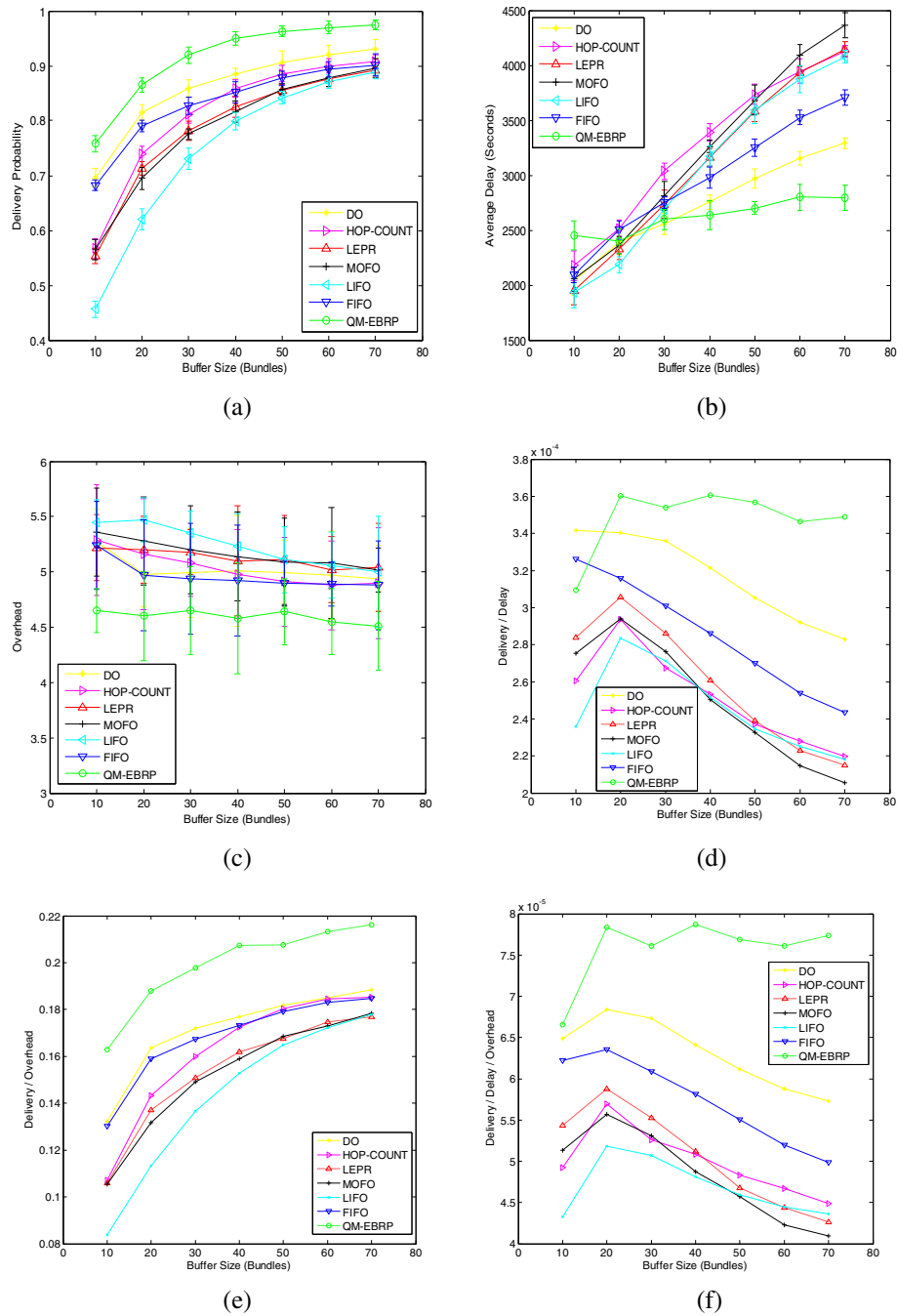


Fig. 10 Network performance under the working day movement model with different node buffer sizes. **a** Delivery probability. **b** Average delay. **c** Overhead. **d** DA. **e** DOR. **f** DAO

QM-EBRP has up to 10 and 36 % improvement, respectively, as compared to DO and FIFO.

5.4 Discussion

Our results suggest that QM-EBRP performs well across all tested scenarios. They confirm QM-EBRP effectively uses the combination of parameters available locally at each node, namely, a node's encounter rate, bundle's

lifetime and number of replicas of a bundle. Indeed, QM-EBRP outperforms other tested policies in terms of both delivery ratio and delay. The reasons that policies such as FIFO, LIFO, LEPR and MOFO perform poorly are their reliance on metrics such as encounter rates or arrival time of a bundle only, which cause these policies to (i) forward bundles that may have insufficient remaining lifetime to be delivered, (ii) drop bundles

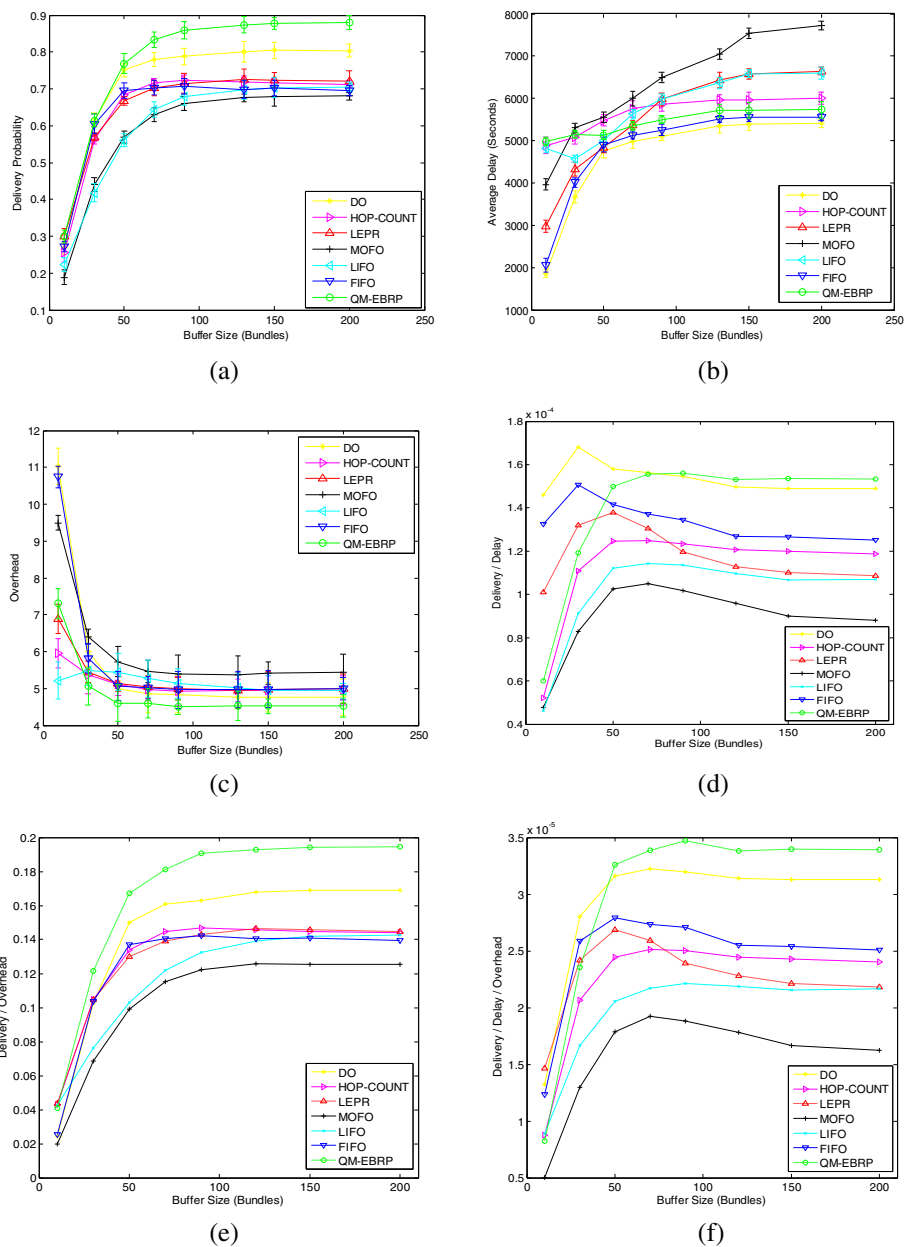


Fig. 11 Network performance under the random mobility model with different node buffer sizes. **a** Delivery probability. **b** Average delay. **c** Overhead. **d** DA. **e** DOR. **f** DAO

with a long remaining lifetime or (iii) drop bundles that have a large number of replicas. In terms of the trade-off between delivery ratio and delay, QM-EBRP outperforms other tested policies. This is because, in the calculation of a bundle's utility, delivery ratio and delay are considered together. However, there are limitations to our approach. Specifically, our approach is not effective in reducing delay under the random mobility model. Recall that QM-EBRP uses nodes' encounter rate in the calculation of a bundle's utility, which helps estimate how likely a bundle will be delivered in the future

and also its expected delay. However, in the random mobility model, a node that has a high rate of encounter rate will not necessarily be reachable in the future.

6 Conclusions

This paper has investigated a novel bundle drop/forward policy for encounter-based quota protocols in DTNs. We proposed a multi-objective function that estimates the delivery ratio and delay of a bundle based on local information that include encounter rate, remaining time to live and number of replicas. This is in contrast to

current queue management policies that require global information. We then calculated the rate of change of both bundle delivery ratio and bundle delivery delay simultaneously. Finally, our proposed policy, QM-EBRP, which uses the resulting multi-objective function, optimizes the global delivery ratio and delay by prioritizing bundles during contacts. We evaluated QM-EBRP over a wide range of scenarios that consider different mobility models and buffer sizes and speeds. Our simulation results showed, under the shortest map-based mobility, QM-EBRP achieved up to 40 % improvement in DAO when nodes have infinite buffer space and up to 30 % when nodes have a limited buffer size over current state-of-the-art policies such as Drop Oldest (DO), Last Input First Output (LIFO), First Input First Output (FIFO), Most FORWARDED first (MOFO), LEast PRObable first (LEPR) and drop greatest HOP-COUNT. Also, under a working day movement mobility model, QM-EBRP performed up to 35 % better in DAO when nodes have different speeds as well as different buffer sizes. Although this scheme outperforms the current state of the art for quota protocols, flooding protocols still suffer from an efficient drop/forward policy. As the future work, we will focus on estimation techniques under flooding-based protocols in order to use accurate information in drop/forward policies.

Competing interests

The authors declare that they have no competing interests.

Received: 9 July 2015 Accepted: 29 February 2016

Published online: 22 March 2016

References

- DTN Research Group, [Online]. Available: <http://www.dtnrg.org>.
- MG Rubinstein, FB Abdesslem, MD de Amorim, SR Cavalcanti, R dos Santos Alves, LHM Costa, OCMB Duarte, MEM Campista, UF Fluminense, Measuring the capacity of in-car to in-car vehicular networks. *Communications Magazine* **47**(11), 128–136 (2009)
- S Iranmanesh, R Raad, K-W Chin, A novel destination-based routing protocol (DBRP) in DTNs, in *International Symposium on Communications and Information Technologies (ISCIT)* (IEEE, Gold Coast, QLD, Australia, 2012)
- SC Nelson, M Bakht, R Kravets, A Harris, Encounter: based routing in DTNs. *SIGMOBILE Mobile Computing and Communications Review* **13**(1), 56–59 (2009)
- SC Nelson, M Bakht, R Kravets, Encounter-based routing in DTNs, in *INFOCOM (IEEE, Rio De Janeiro, Brazil, 2009)*
- S-Y Ni, Y-C Tseng, Y-S Chen, J-P Sheu, The broadcast storm problem in a mobile ad hoc network. *Wirel. Netw.* **8**(2), 153–167 (1999)
- EPC Jones, L Li, JK Schmidtke, PAS Ward, Practical routing in delay-tolerant networks. *Transactions on Mobile Computing* **6**(8), 943–959 (2007)
- P Juang, H Oki, Y Wang, M Martonosi, LS Peh, D Rubenstein, Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems* **30**(5), 96–107 (2002)
- V Erramilli, M Crovella, Forwarding in opportunistic networks with resource constraints, in *CHANTS '08 Proceedings of the third ACM workshop on Challenged networks* (ACM, San Francisco, California, USA, 2008)
- T Spyropoulos, K Psounis, CS Raghavendra, Spray and focus: efficient mobility-assisted routing for heterogeneous and correlated mobility, in *Fifth Annual International Conference on Pervasive Computing and Communications Workshops* (IEEE, White Plains, New York, USA, 2007)
- S. Kapadia, B. Krishnamachari, L. Zhang, "Mobile Ad-Hoc Networks: Protocol Design", Chapter 26: 'Data Delivery in Delay Tolerant Networks: A Survey. InTech, 565–578 (2011).
- J Miao, O Hasan, SB Mokhtar, L Brunie, A self-regulating protocol for efficient routing in mobile delay tolerant networks, in *6th International Conference on Digital Ecosystems Technologies (DEST)* (IEEE, Campion d'Italia, Italy, 2012)
- MAT Prodhan, R Das, MH Kabir, GC Shoja, Probabilistic quota based adaptive routing in Opportunistic Networks, in *Pacific Rim Conference on Communications, Computers and Signal Processing (PacRim)* (IEEE, Victoria, BC, Canada, 2011)
- S-C Lo, W-R Liou, Dynamic quota-based routing in delay-tolerant networks, in *Vehicular Technology Conference (VTC)* (IEEE, Yokohama, Japan, 2012)
- X Zhang, G Neglia, J Kurose, D Towsley, Performance modeling of epidemic routing. *Computer Networks: The International Journal of Computer and Telecommunications Networking* **51**(10), 2867–2891 (2007)
- D Kim, H Park, I Yeom, Minimizing the impact of buffer overflow in dtn, in *Proceedings International Conference on Future Internet Technologies (CFIT)*, 2008
- A Lindgren, KS Phanse, Evaluation of queueing policies and forwarding strategies for routing in intermittently connected networks, in *First International Conference on Communication System Software and Middleware* (IEEE, New Delhi, India, 2006)
- A Lindgren, A Doria, O Schelen, Probabilistic routing in intermittently connected networks. *SIGMOBILE Mobile Computing and Communications Review* **7**(3), 19–20 (2003)
- A Elwhishi, P-H Ho, K Naik, B Shihada, A novel message scheduling framework for delay tolerant networks routing. *Transactions on Parallel and Distributed Systems* **24**(5), 871–880 (2013)
- A Krifa, C Barakat, T Spyropoulos, Optimal buffer management policies for delay tolerant networks, in *5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks SECON '08* (IEEE, San Francisco, CA, USA, 2008)
- A Krifa, C Barakat, T Spyropoulos, Message drop and scheduling in DTNs: theory and practice. *Transactions on Mobile Computing* **11**(9), 1470–1483 (2012)
- A Balasubramanian, B Levine, A Venkataramani, DTN routing as a resource allocation problem. *SIGCOMM Computer Communication Review* **37**(4), 373–384 (2007)
- Y Li, M Qian, D Jin, L Su, L Zeng, Adaptive optimal buffer management policies for realistic DTN, in *Global Telecommunications Conference GLOBECOM* (IEEE, Honolulu, Hawaii, USA, 2009)
- S Rashid, Q Ayub, AH Abdullah, Reactive weight based buffer management policy for DTN routing protocols. *Wirel. Pers. Commun.* **80**(3), 993–1010 (2015)
- B Burns, O Brock, BN Levine, MV routing and capacity building in disruption tolerant networks, in *INFOCOM, Proceeding of 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, 2005
- Q Ayub, S Rashid, T-Drop: an optimal buffer management policy to improve QOS in DTN routing protocols. *Journal of Computing* **2**, 10 (2010)
- D Pan, Z Ruan, N Zhou, X Liu, Z Song, A comprehensive-integrated buffer management strategy for opportunistic networks. *EURASIP J. Wirel. Commun. Netw.* 103–113 (2013)
- JF Naves, IM Moraes, C Albuquerque, LPS and LRF: efficient buffer management policies for delay and disruption tolerant networks, in *37th Conference on Local Computer Networks (LCN)* (IEEE, Clearwater, FL, USA, 2012)
- G Fathima, RSD Wahidabanu, Buffer management for preferential delivery in opportunistic delay tolerant networks. *International Journal of Wireless & Mobile Networks* **3**(5), 15 (2011)
- C Rohner, F Bjurefors, P Gunningberg, L McNamara, E Nordstrom, Making the most of your contacts: transfer ordering in data-centric opportunistic networks, in *MobiOpp '12 Proceedings of the third ACM international workshop on Mobile Opportunistic Networks* (ACM, Zurich, Switzerland, 2012)
- L Yin, HM Lu, Y da Cao, JM Gao, Buffer scheduling policy in DTN routing protocols, in *2nd International Conference on Future Computer and Communication (ICFCC)* (IEEE, Wuhan, China, 2010)
- D Pan, W Cao, H Zhang, M Lin, Buffer management and hybrid probability choice routing for packet delivery in opportunistic networks. *Math. Probl. Eng.* **2012** 1–14 (2012)
- Y Liu, J Wang, S Zhang, H Zhou, A buffer management scheme based on message transmission status in delay tolerant networks, in *Global Telecommunications Conference (GLOBECOM)* (IEEE, Houston, TX, USA, 2011)
- K Shin, S Kim, Enhanced buffer management policy that utilizes message properties for delay-tolerant networks. *Communications* **5**(6), 753–759 (2011)

35. R Ramanathan, R Hansen, P Basu, R Rosales-Hain, R Krishnan, Prioritized epidemic routing for opportunistic networks, in *MobiOpp '07 Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking* (ACM, San Juan, Puerto Rico, 2007)
36. MJ Khabbaz, CM Assi, WF Fawaz, Disruption-tolerant networking: a comprehensive survey on recent developments and persisting challenges. *Communications Surveys & Tutorials* **14**(2), 607–640 (2012)
37. Y Zhu, B Xu, X Shi, Y Wang, A survey of social-based routing in delay tolerant networks: positive and negative social effects. *Communications Surveys & Tutorials* **15**(1), 387–401 (2013)
38. JA Dias, JN Isento, VNGJ Soares, JJPC Rodrigues, Impact of scheduling and dropping policies on the performance of vehicular delay-tolerant networks, in *International Conference on Communications (ICC)* (IEEE, Kyoto, Japan, 2011)
39. VNGJ Soares, F Farahmand, JJPC Rodrigues, Performance analysis of scheduling and dropping policies in vehicular delay-tolerant networks. *International Journal on Advances in Internet Technology - IARIA* **3**(1), 137–145 (2010)
40. M Ke, Y Nenghai, L Bin, A new packet dropping policy in delay tolerant network, in *12th IEEE International Conference on Communication Technology (ICCT)* (IEEE, Nanjing, China, 2010)
41. S Rashid, Q Ayub, MSM Zahid, AH Abdullah, Impact of mobility models on DLA (drop largest) optimized DTN epidemic routing protocol. *Int. J. Comput. Appl.* **18**(5), 1–7 (2011)
42. A Vahdat, D Becker, *Epidemic routing for partially connected ad hoc networks* (Duke University, Durham NC, 2000)
43. T Spyropoulos, K Psounis, CS Raghavendra, Spray and wait: an efficient routing scheme for intermittently connected mobile networks, in *WDTN '05 Proceedings of the SIGCOMM workshop on Delay-tolerant networking* (ACM, Philadelphia, Pennsylvania, USA, 2005)
44. S Rashid, Q Ayub, MSM Zahid, SH Abdullah, E-DROP: an effective drop buffer management policy for DTN routing protocols. *Int. J. Comput. Appl.* **13**(7), 8–13 (2011)
45. S Rashid, AH Abdullah, MSM Zahid, Q Ayub, Mean drop an effectual buffer management policy for delay tolerant network. *Eur. J. Sci. Res.* **70**(3), 396–407 (2012)
46. Q Ayub, S Rashid, MSM Zahid, Buffer scheduling policy for opportunistic networks. *International Journal of Scientific & Engineering Research (IJSER)* **2**, 7 (2011)
47. HEL Amornsin, Heuristic congestion control for message deletion in delay tolerant network. In *Smart Spaces and Next Generation Wired/Wireless Networking*, in *Third Conference on Smart Spaces and 10th International Conference*, 2010
48. S Rashid, Q Ayub, MSM Zahid, AH Abdullah, Message drop control buffer management policy for DTN routing protocols. *Wirel. Pers. Commun.* **72**(1), 653–669 (2013)
49. L Yun, C Xinjian, L Qilie, Y Xianohu, A novel congestion control strategy in delay tolerant networks, in *Second International Conference on Future Networks* (IEEE, Sanya, Hainan, 2010)
50. R Groenevelt, P Nain, G Koole, Message delay in MANET. *SIGMETRICS Performance Evaluation Review* **33**(1), 412–413 (2005)
51. AP Silva, S Burleigh, CM Hirata, K Obraczka, A survey on congestion control for delay and disruption tolerant. *Ad Hoc Netw.* **25**(2), 480–494 (2014)
52. E Coe, C Rachavendra, Token based congestion control for DTN, in *Aerospace Conference* (IEEE, Big Sky, MT, 2010)
53. N Thompson, R Kravets, Understanding and controlling congestion in DTNs. *SIGMOBILE Mobile Computing and Communications Review* **13**(3), 42–45 (2009)
54. J Lakkakorpi, M Pitkanen, J Ott, Using buffer space advertisements to avoid congestion in mobile opportunistic DTNs, in *WWIC'11 Proceedings of the 9th IFIP TC 6 international conference on Wired/wireless internet communications* (SPRINGER, Vilanova i la Geltru, Spain, 2011)
55. M Seligman, K Fall, P Mundur, Storage routing for DTN congestion control. *Wireless Communications & Mobile Computing - Wireless Ad Hoc and Sensor Network* **7**(10), 1183–1196 (2007)
56. P Hui, J Crowcroft, E Yoneki, BUBBLE rap: social-based forwarding in delay-tolerant networks. *Transactions on Mobile Computing* **10**(11), 1576–1589 (2011)
57. MC Gonzalez, CA Hidalgo, AL Barabasi, Understanding individual human mobility patterns. *Nature* **453**, 779–782 (2008)
58. A. Keranen, J. Ott, *Increasing reality for DTN protocol simulations* (Helsinki University of Technology, Helsinki, Finland). (2007)
59. F Bai, N Sadagopan, A Helmy, IMPORTANT: a framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks, in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications* (IEEE, San Francisco, USA, 2003)
60. T Camp, J Boleng, V Davies, A survey of mobility models for ad hoc network research. *Wirel. Commun. Mob. Comput.* **2**(5), 483–502 (2002)
61. S Iranmanesh, K-W Chin, Mobility based routing protocols for semi-predictable disruption tolerant networks. *Int. J. Wireless Inf. Networks* **22**(2), 138–146 (2015)
62. T Karagiannis, J-YL Boudec, M Vojnovic, Power law and exponential decay of intercontact times between mobile devices. *Transactions on Mobile Computing* **9**(10), 1377–1390 (2010)
63. T Spyropoulos, K Psounis, CS Raghavendra, Performance analysis of mobility-assisted routing, in *MobiHoc '06 Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing* (ACM, Florence, Italy, 2006)
64. M McNett, GM Voelker, Access and mobility of wireless PDA users. *SIGMOBILE Mobile Computing and Communications Review* **9**(2), 40–55 (2005)
65. J Krumm, E Horvitz, *The Microsoft Multi Person Location Survey* (Microsoft Research TechReport, 2005)
66. N Eagle, A Pentland, *CRAWDAD Data Set Mit/Reality (v. 2005-07-01)*, 2005
67. N Eagle, A Pentland, Reality mining: sensing complex social systems. *Journal of Personal Ubiquitous Computing* **10**(4), 255–268 (2006)
68. A Chaintreau, P Hui, J Crowcroft, C Diot, R Gass, J Scott, Impact of human mobility on opportunistic forwarding algorithms. *Transactions on Mobile Computing* **6**(6), 606–620 (2007)
69. J Scott, R Gass, J Crowcroft, P Hui, C Diot, A Chaintreau, *CRAWDAD Data Set Cambridge/Haggle (v. 2006-01-31)*, 2006
70. J Scott, R Gass, J Crowcroft, P Hui, C Diot, A Chaintreau, *CRAWDAD Trace Cambridge/Haggle/Imote/Infocom (v. 2006-01-31)*, 2006
71. M Taboga, *Lectures on probability theory and mathematical statistics*, Second ed. Amazon CreateSpace **13**, 978–1480215238 (2012)
72. A Keranen, J Ott, T Karkkainen, The ONE simulator for DTN protocol evaluation, in *Simutools '09 Proceedings of the 2nd International Conference on Simulation Tools and Techniques* (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, Rome, Italy, 2009)
73. F Ekman, A Keranen, J Karvo, J Ott, Working day movement model, in *MobilityModels '08 Proceedings of the 1st ACM SIGMOBILE workshop on Mobility models* (ACM, Hong Kong, Hong Kong, China, 2008)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com